

Module 5:VPC AND PEERING

SUBMITTED BY :-HITESH CHAUHAN

COURSES OFFERED:ADVANCED CLOUD COMPUTING AND DEVELOPS

VPC AND PEERING ASSIGNMENT

You work for XYZ Corporation and based on the expansion requirements of your corporation you have been asked to create and set up a distinct Amazon VPC for the production and development team. You are expected to perform the following tasks for the respective VPCs.

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.

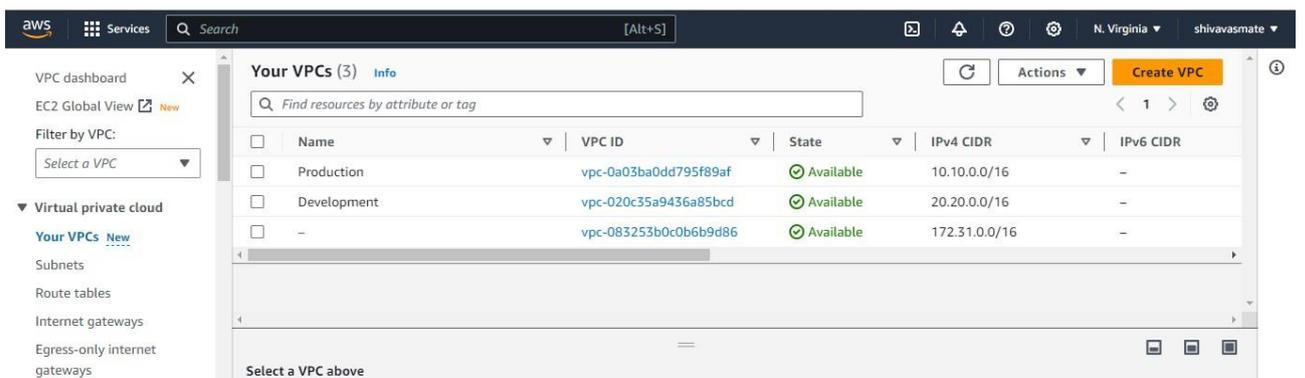
Development Network:

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.

Ans:

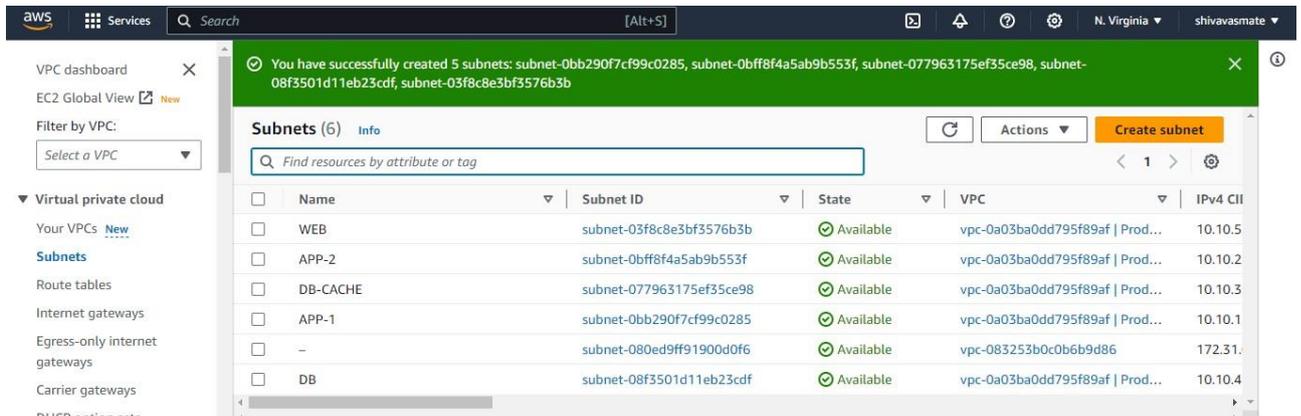
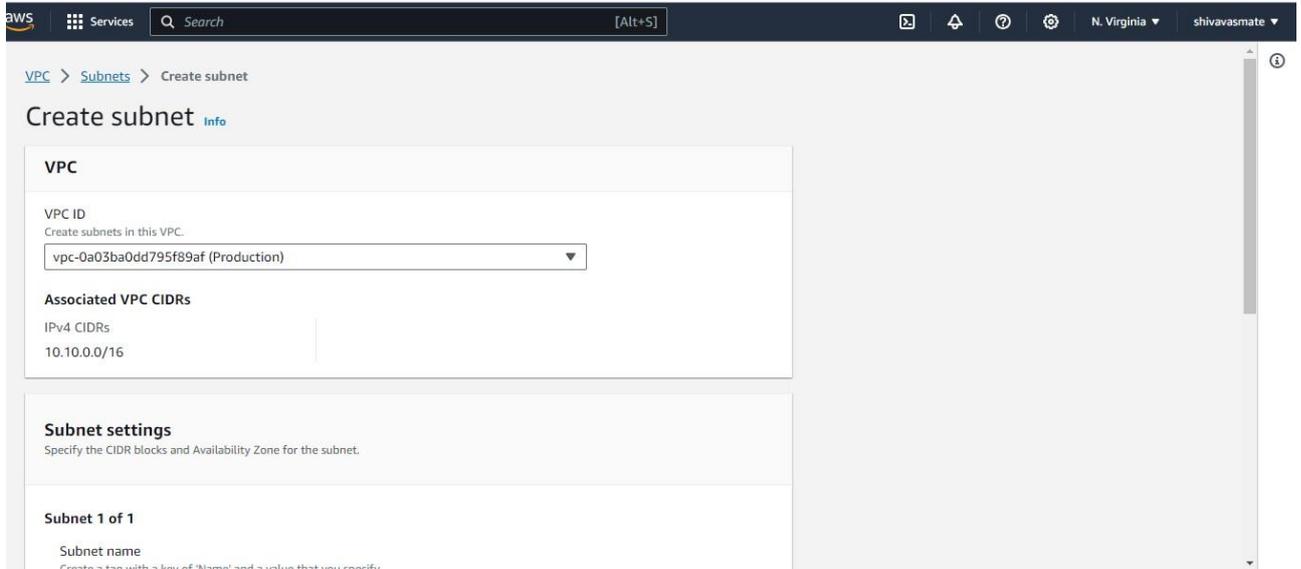
For Production Network:

1. Open up the AWS Management Console
2. Check for the region [us-east-1(N. Virginia)]
3. Now create two VPC namely 1. Production and 2. Development

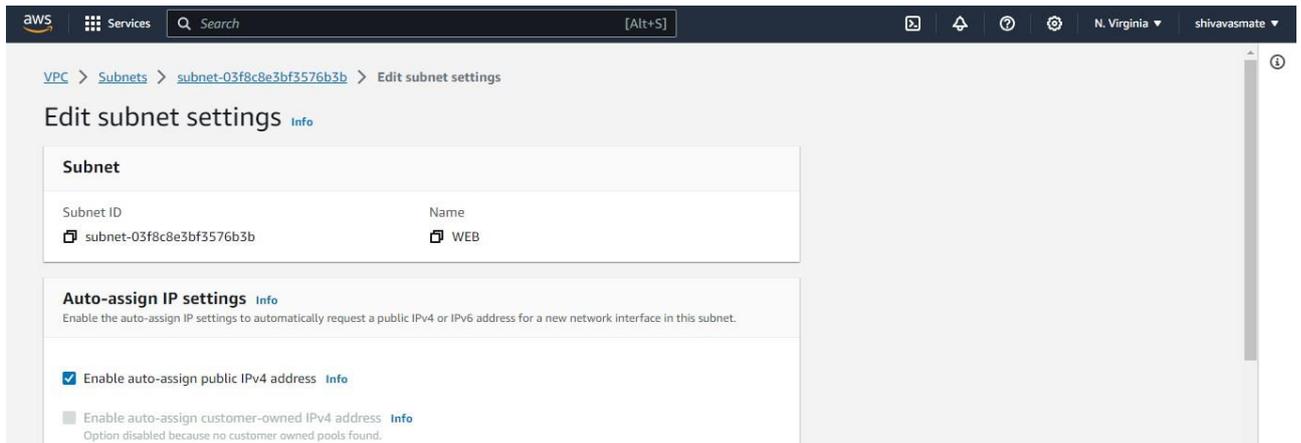


4. Create five subnets in the production vpc namely

a) APP-1 b) APP-2 c) DB d) DB-CACHE and e)WEB



5. Making WEB as subnet Public one by Auto-assign IP setting.



6. Create internet gateway with a name Production-IGW

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

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

Production-IGW

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key: Name Value - optional: Production-IGW [Remove]

7. Attach this Production-IGW to Production VPC.

Attach to VPC (igw-0a640877ab149cb1f) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

vpc-0a03ba0dd795f89af

▶ AWS Command Line Interface command

Cancel **Attach internet gateway**

EC2 Global View [New](#)

Filter by VPC:

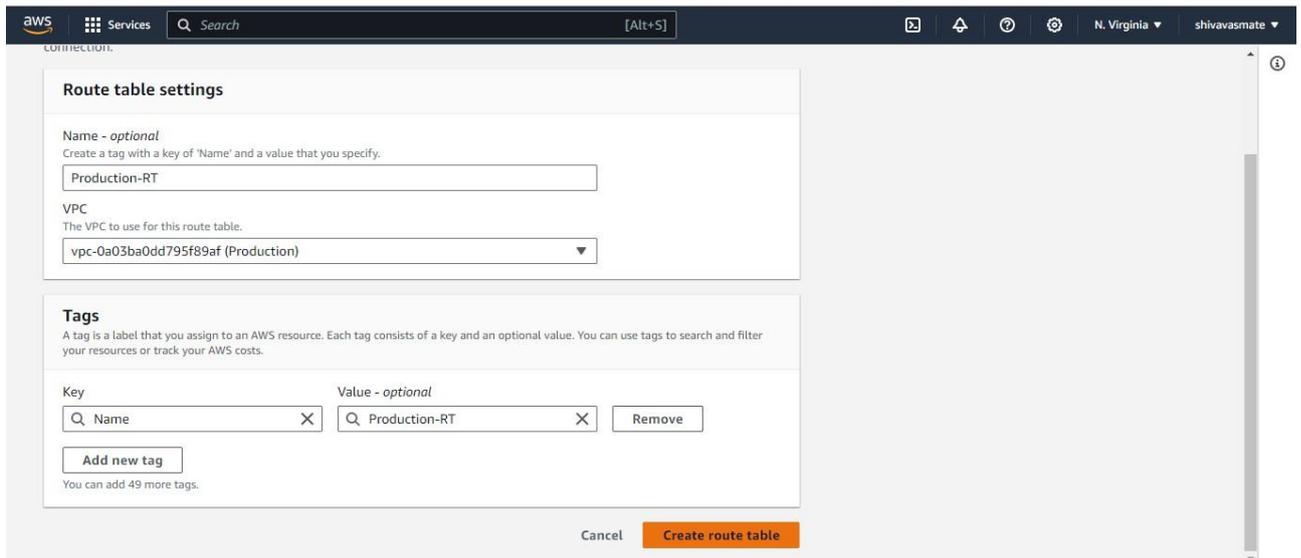
Virtual private cloud

Your VPCs [New](#)

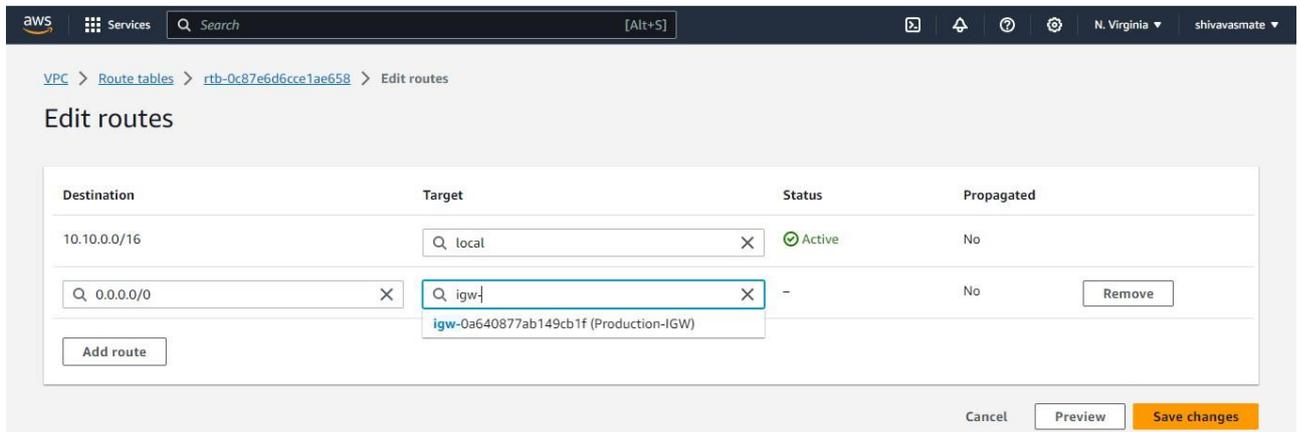
Subnets

	Name	Internet gateway ID	State	VPC ID	Owner
<input checked="" type="checkbox"/>	Production-IGW	igw-0a640877ab149cb1f	Attached	vpc-0a03ba0dd795f89af Production	29708108
<input type="checkbox"/>	-	igw-0ff037c6413a4441a	Attached	vpc-083253b0c0b6b9d86	29708108

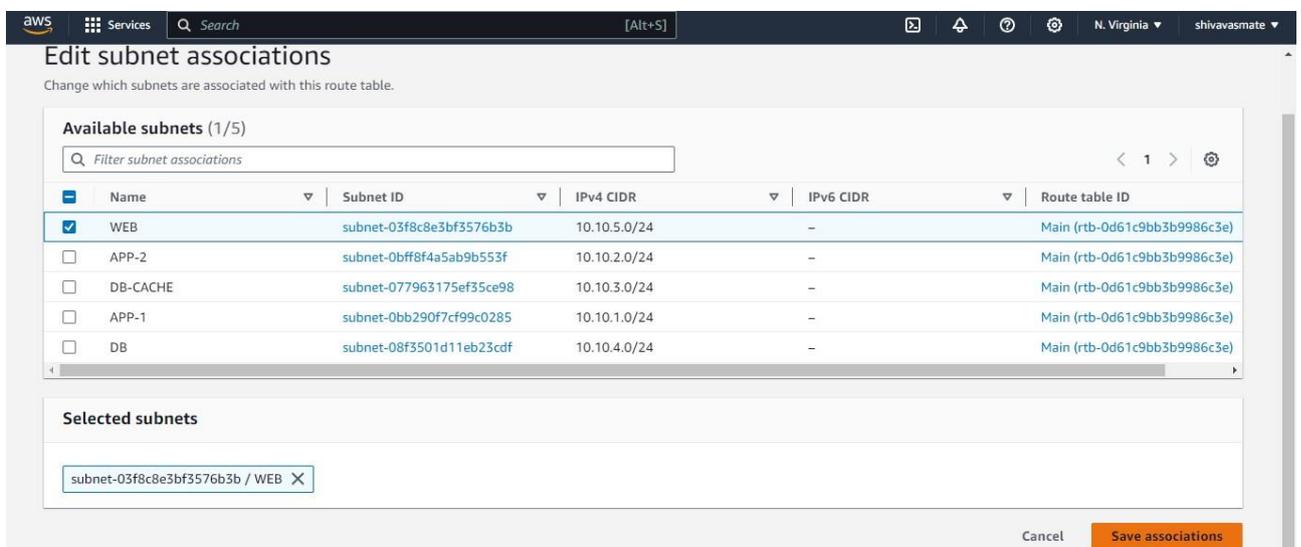
8. Create Route table with the name production-RT in VPC of Production.



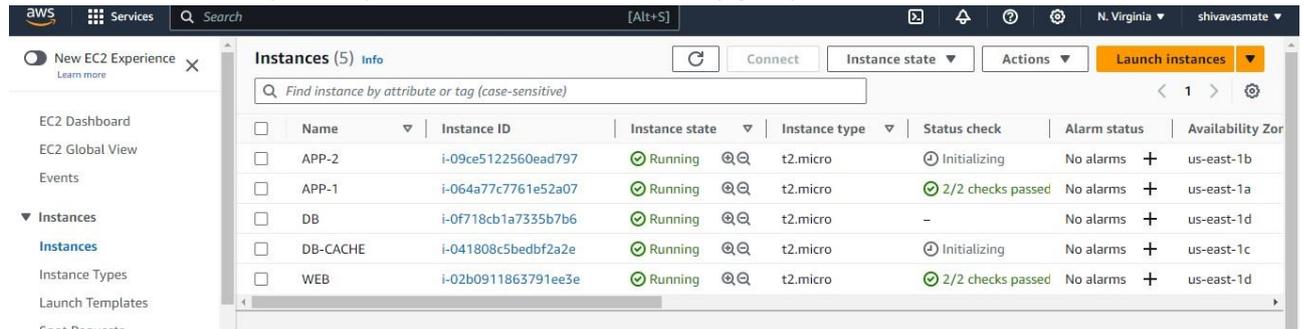
9. Edit routes with destination as internet (0.0.0.0/0) to target internet gateway of production (Production-IGW).



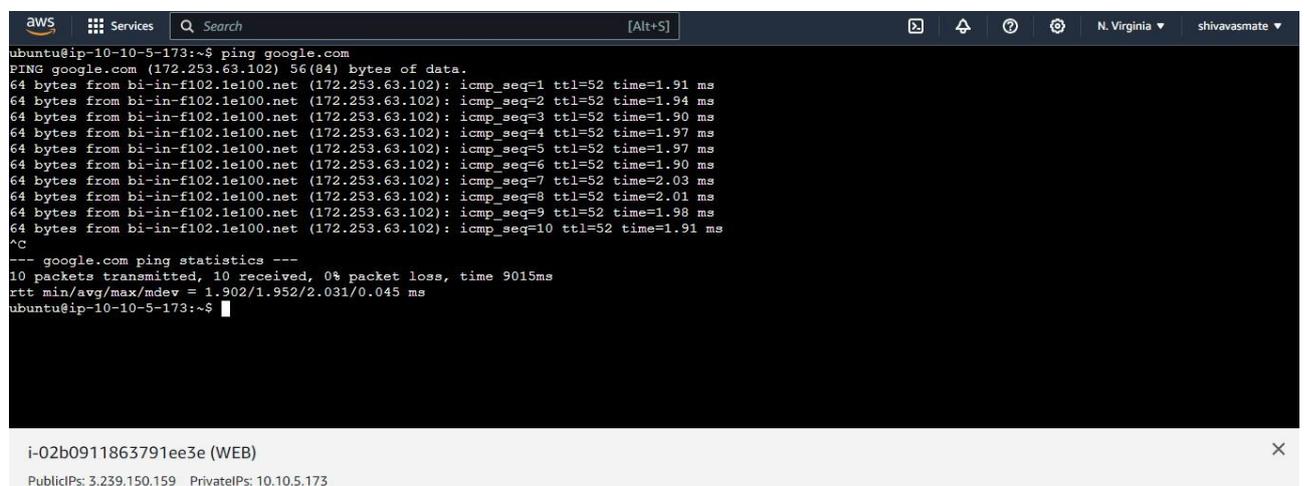
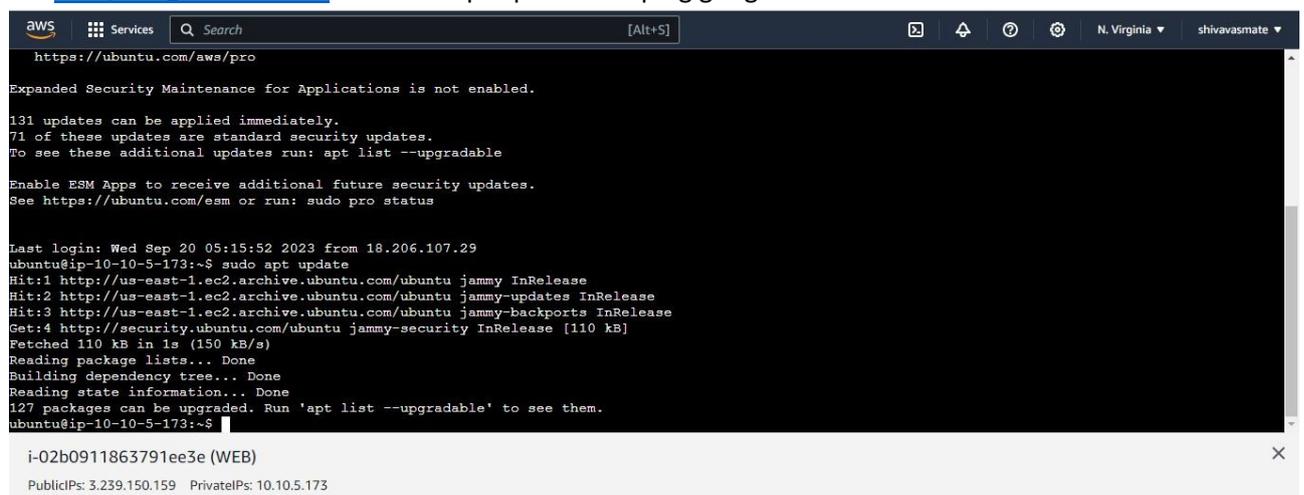
10. To get internet to public subnet (WEB) make subnet Association with WEB.



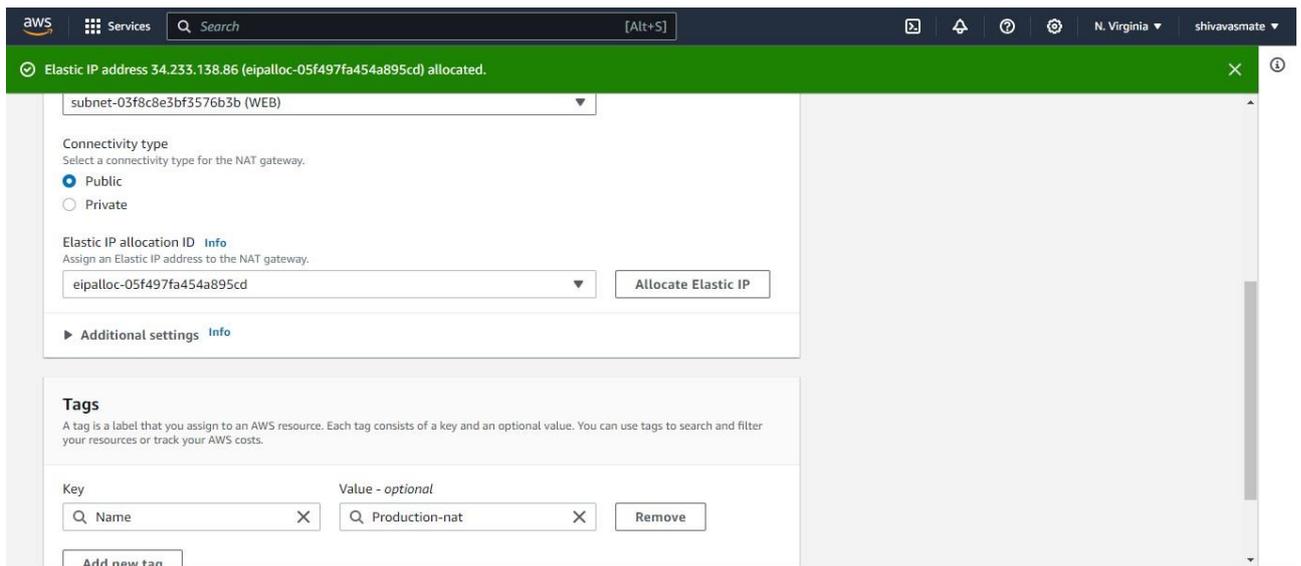
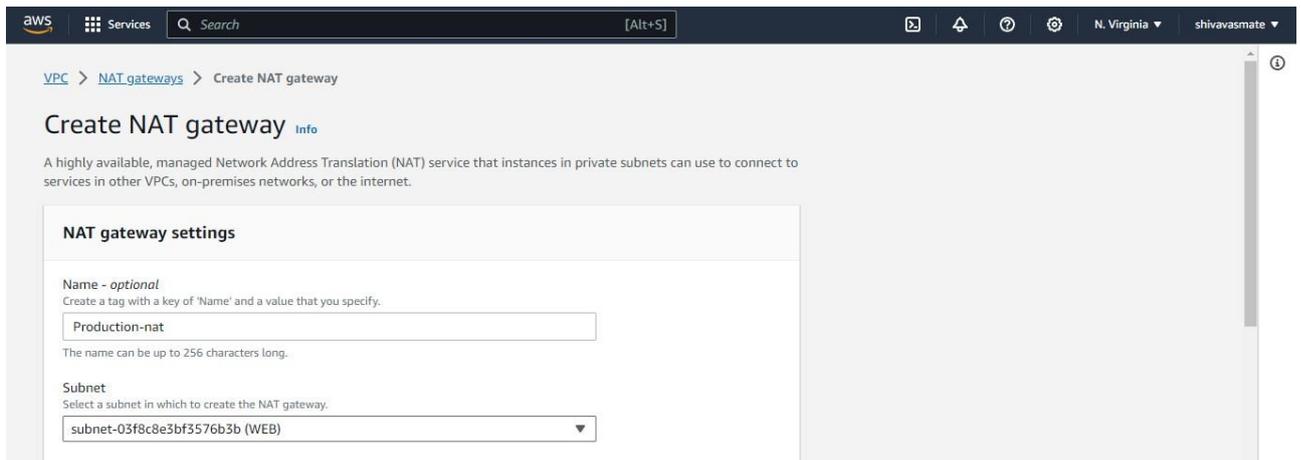
11. Now launch five EC2 instances in the Production vpc namely
a) APP-1 b) APP-2 c) DB d) DB-CACHE and e) WEB with all ubuntu instances in their respectively subnets with default security groups.



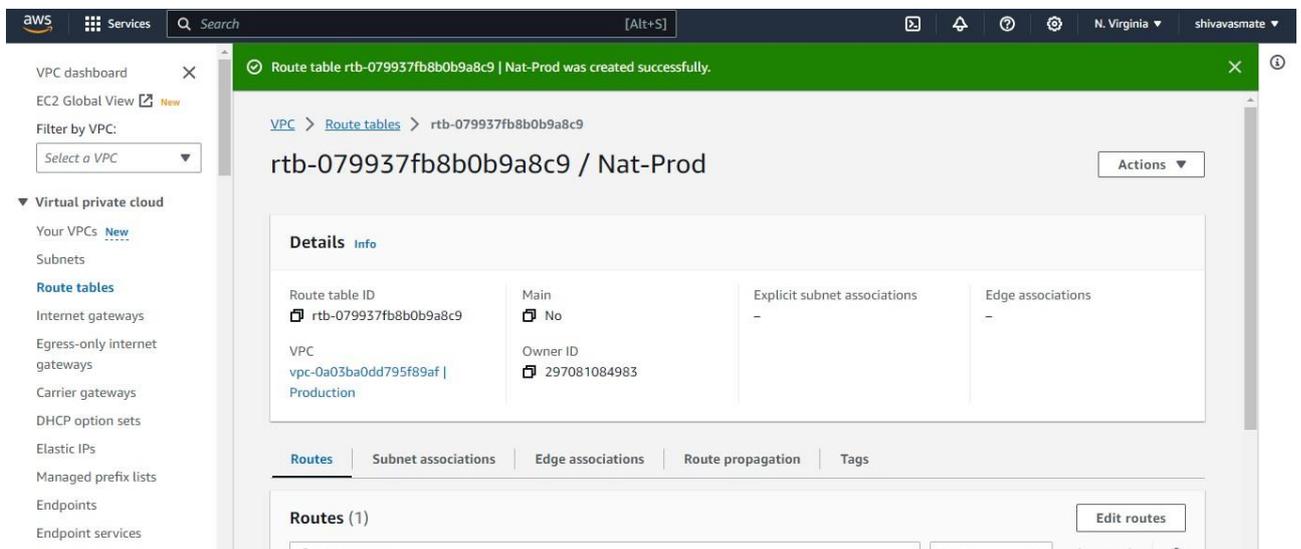
12. Now check the internet connecting in the public subnet (WEB). So connect instance and update with commands `sudo nano test.pem`, `sudo chmod 400 test.pem`, `sudo ssh -i ubuntu test.pem@10.10.5.173` and `sudo apt update` and `ping google.com`

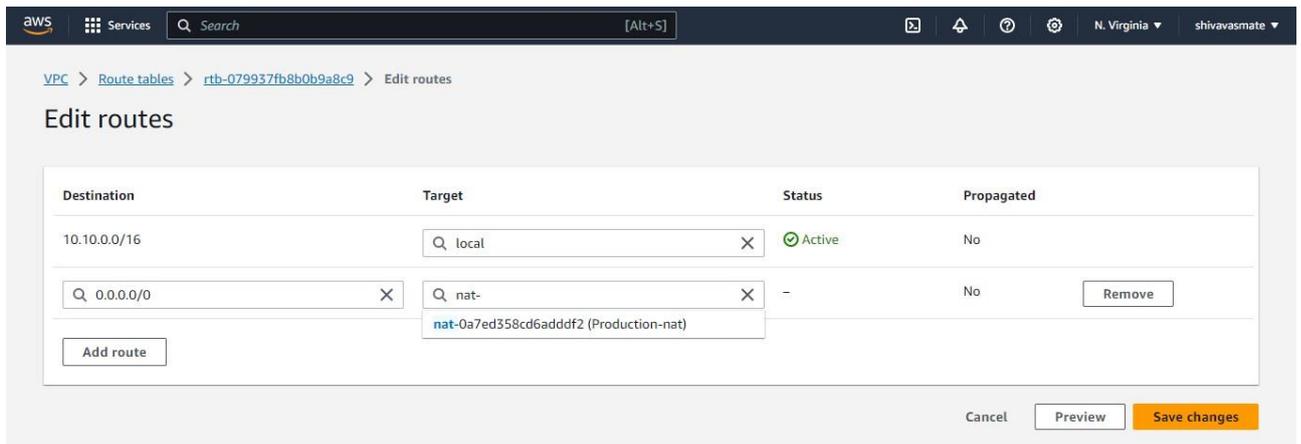


13. As per the question allow db-cache instance and app1 subnet to send internet requests. For that create NAT gateway with the name Production-nat in the WEB subnet and connectivity type public with a elastic IP allocation to it.

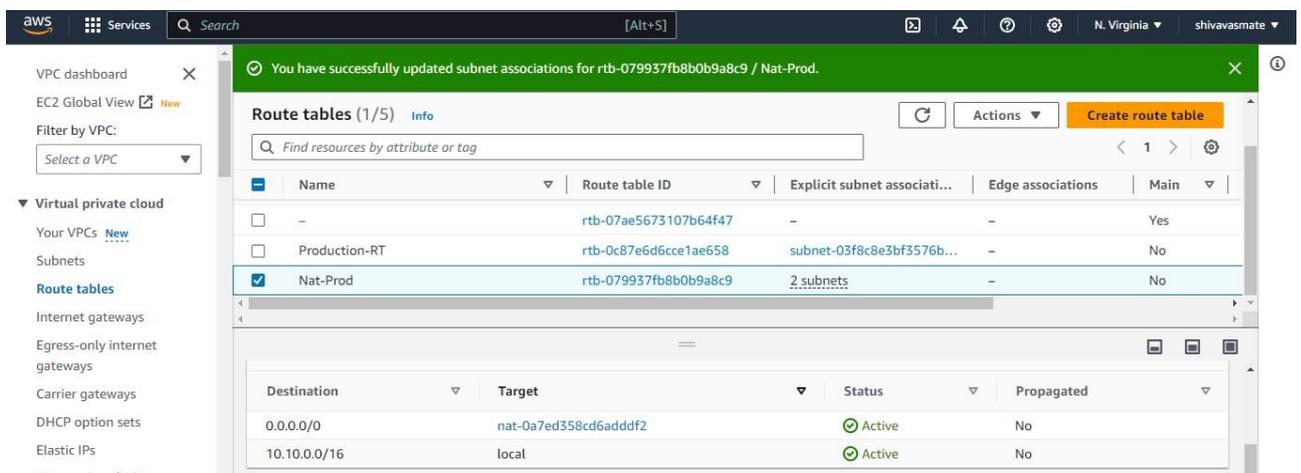
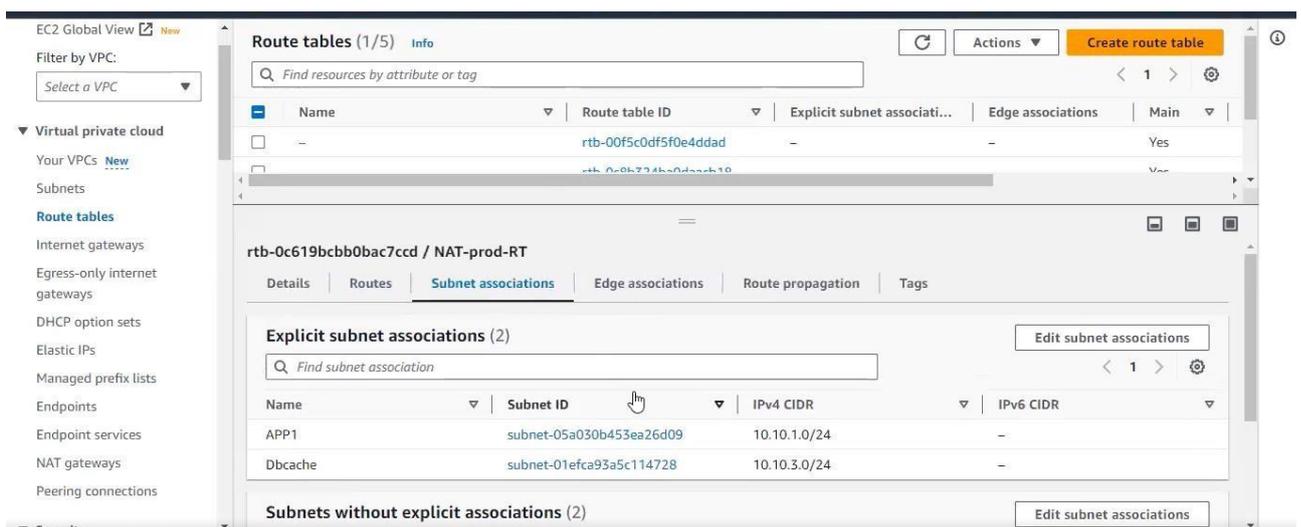


14. Create another route table with a name Nat-Prod and edit route for make connection to public (WEB) subnet to private subnets (APP-1 and DB-CACHE to pass internet. As destination 0.0.0.0/0 to target Nat gateway of Production-Nat





15. Make subnet association with APP-1 and DB-CACHE for internet request.



16. To check internet request from public subnet of WEB instance to the APP-1 and DB-CACHE instances

Now check the internet connecting in the public subnet (WEB). So connect instance and update with commands `sudo nano test.pem`, `sudo chmod 400 test.pem`, `sudo ssh -i ubuntu test.pem@10.10.1.178` and `sudo apt update` and ping google.com

```

--- amazon.com ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20035ms
rtt min/avg/max/mdev = 10.102/10.190/10.406/0.069 ms
ubuntu@ip-10-10-5-230:~$ sudo nano test.pem
ubuntu@ip-10-10-5-230:~$ sudo chmod 400 test.pem
ubuntu@ip-10-10-5-230:~$ sudo ssh -i test.pem ubuntu@10.10.1.178

```

```

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

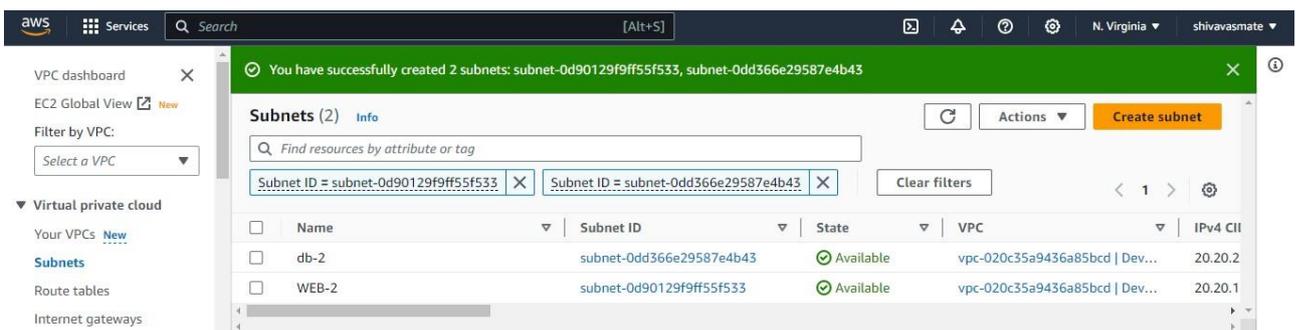
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-1-178:~$ ping google.com
PING google.com (142.250.190.110) 56(84) bytes of data:
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=1 ttl=113 time=18.3 ms
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=2 ttl=113 time=17.8 ms
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=3 ttl=113 time=17.9 ms
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=4 ttl=113 time=17.9 ms
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=5 ttl=113 time=17.9 ms
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=6 ttl=113 time=18.0 ms
64 bytes from ord37s35-in-f14.1e100.net (142.250.190.110): icmp_seq=7 ttl=113 time=17.9 ms

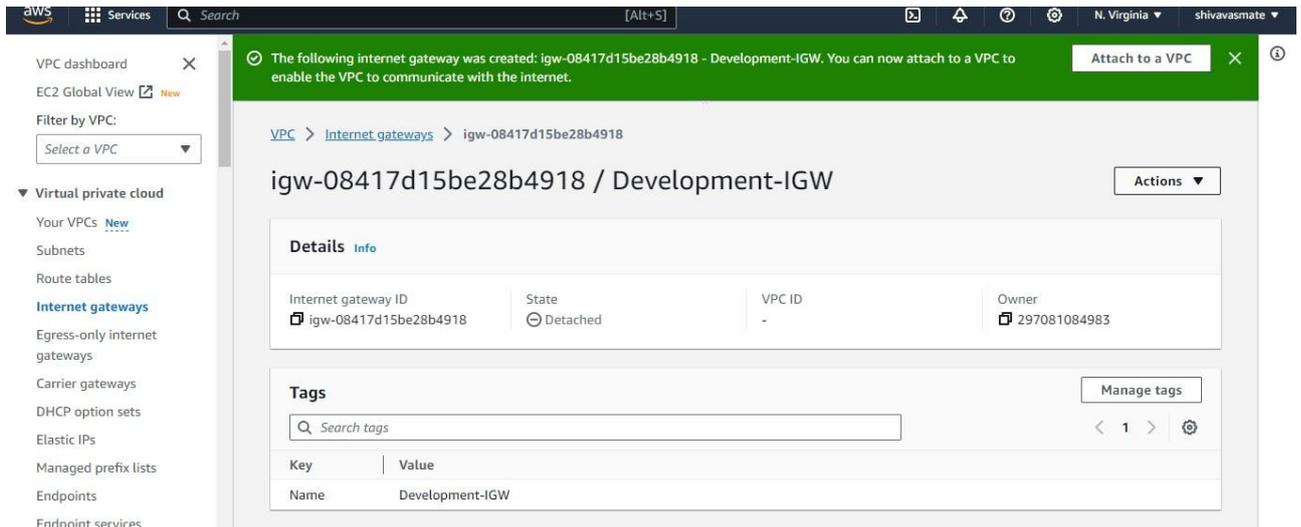
```

For Development Network:

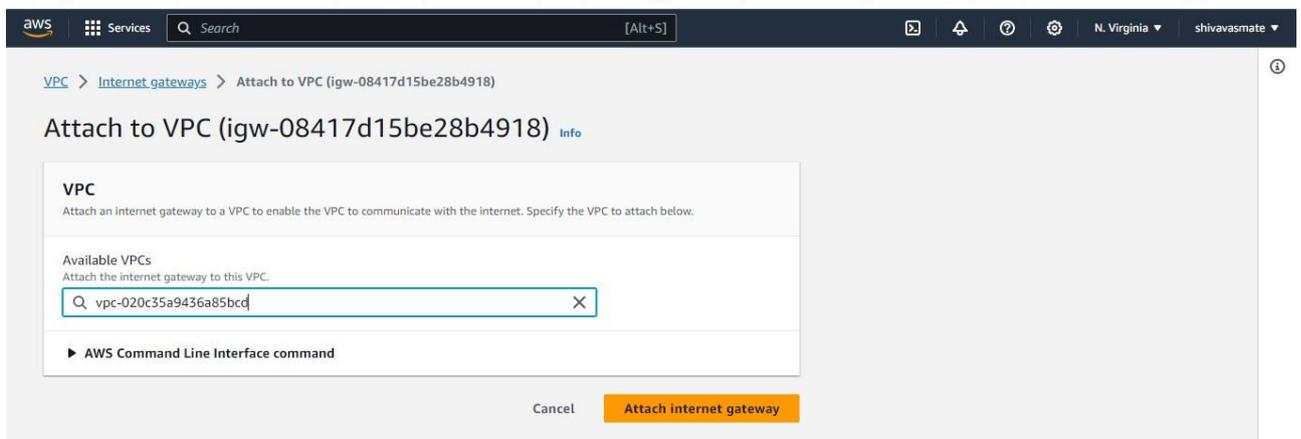
1. Already VPC for development network created.
2. Now create two subnet namely a) WEB-2 and b) DB-2 in the development vpc.



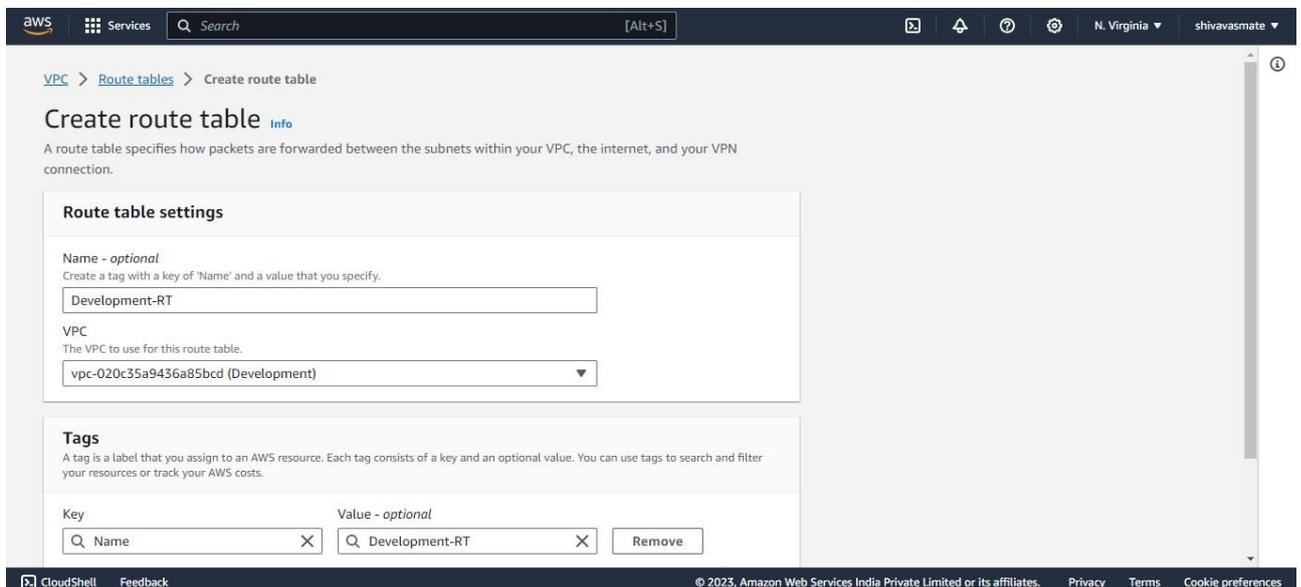
3. Create internet gateway with a name Development-IGW



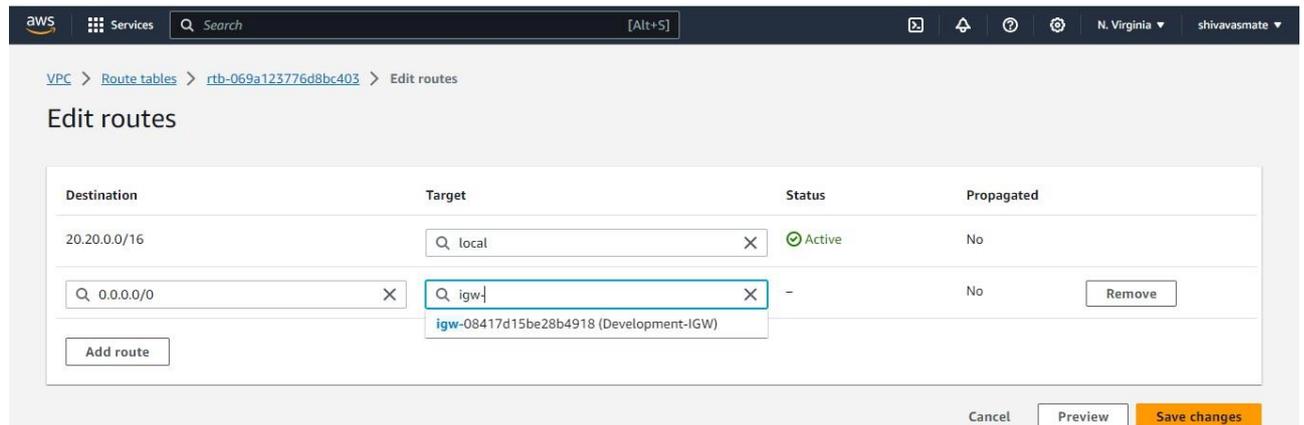
4. Attach this Development-IGW to Development VPC.



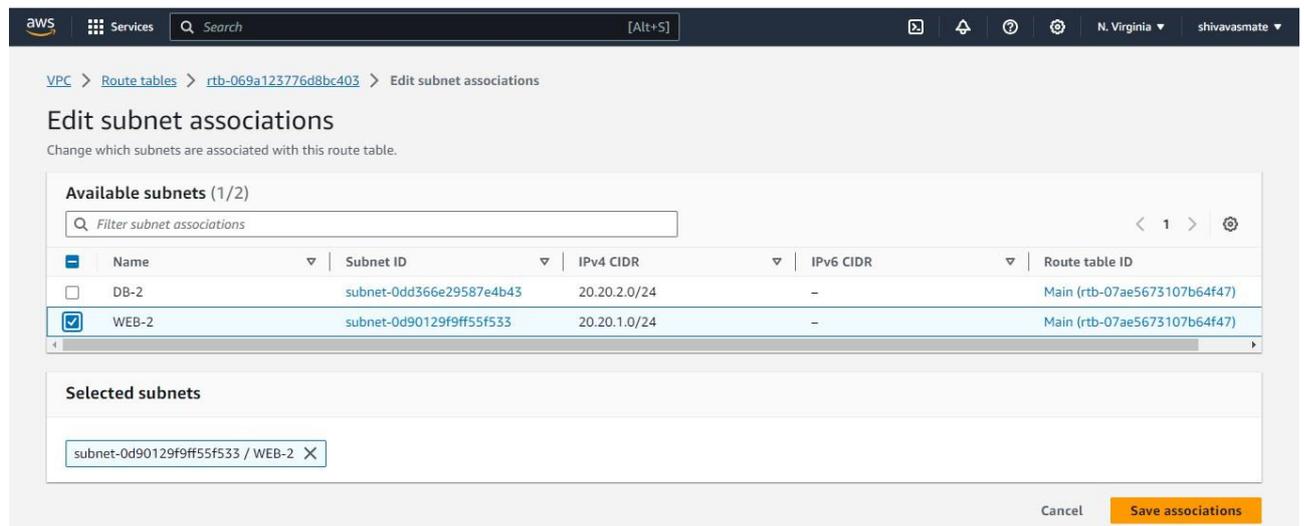
5. Create Route table with the name Development-RT in VPC of development.



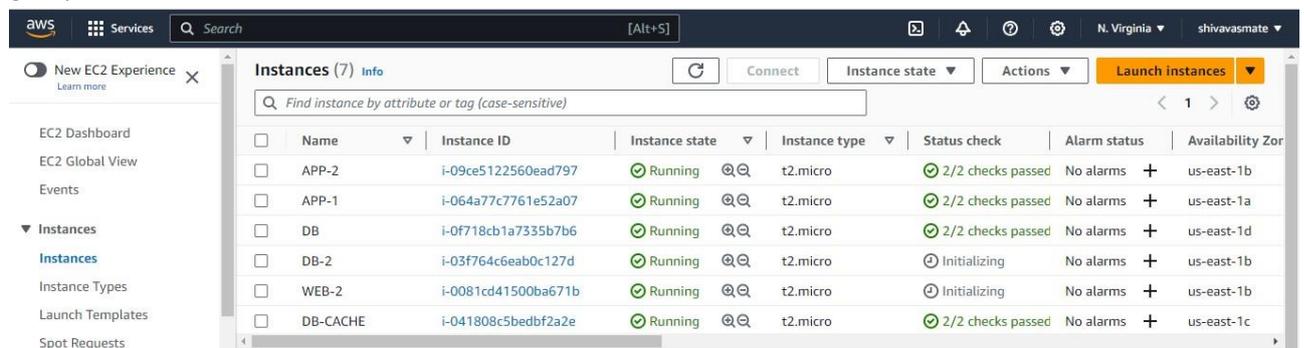
6. Edit routes with destination as internet (0.0.0.0/0) to target internet gateway of development (DevelopmentIGW).



7. To get internet to public subnet (WEB-2) make subnet Association with WEB.



8. Now launch five EC2 instances in the Development vpc namely
 a) WEB-2 b) DB-2 with all ubuntu instances in their respectively subnets with default security groups.



9. Now check the internet connecting in the public subnet (WEB-2). Sudo apt update and ping google.com

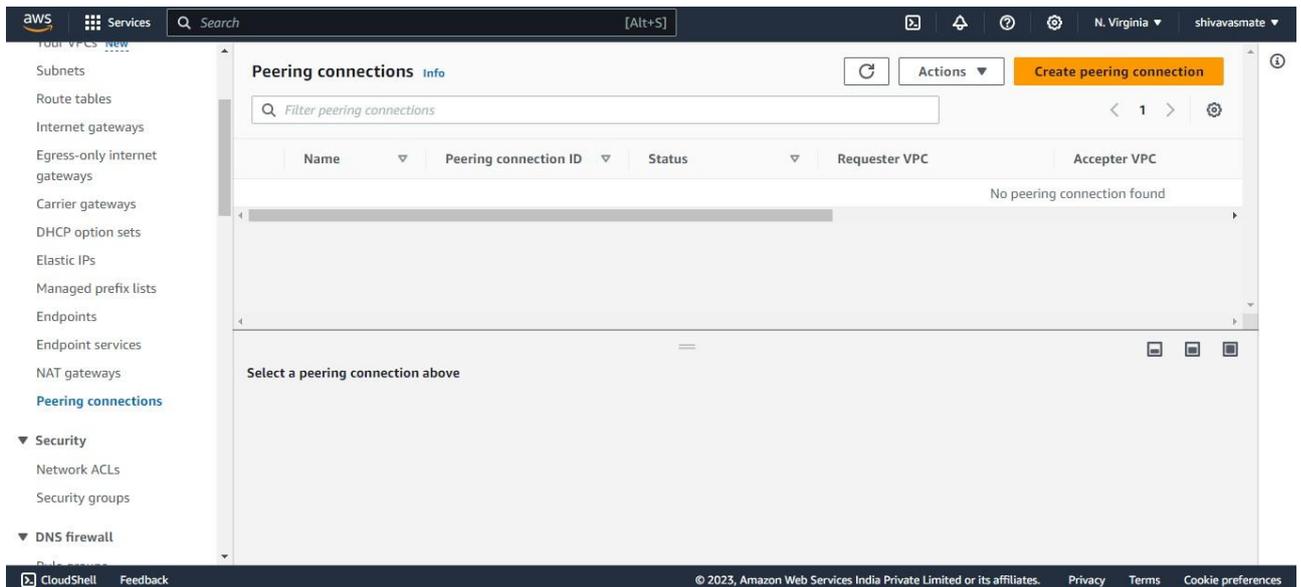
```
aws
Services Search [Alt+S]
N. Virginia shivavasmate
ubuntu@ip-20-20-1-94:~$ ping google.com
PING google.com (142.251.167.101) 56(84) bytes of data:
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=1 ttl=100 time=5.50 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=2 ttl=100 time=1.78 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=3 ttl=100 time=1.82 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=4 ttl=100 time=1.84 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=5 ttl=100 time=1.72 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=6 ttl=100 time=1.80 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=7 ttl=100 time=1.82 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=8 ttl=100 time=1.79 ms
64 bytes from ww-in-f101.1e100.net (142.251.167.101): icmp_seq=9 ttl=100 time=1.74 ms

i-0d9377012c839ada3 (web-2)
PublicIPs: 3.95.13.13 PrivateIPs: 20.20.1.94

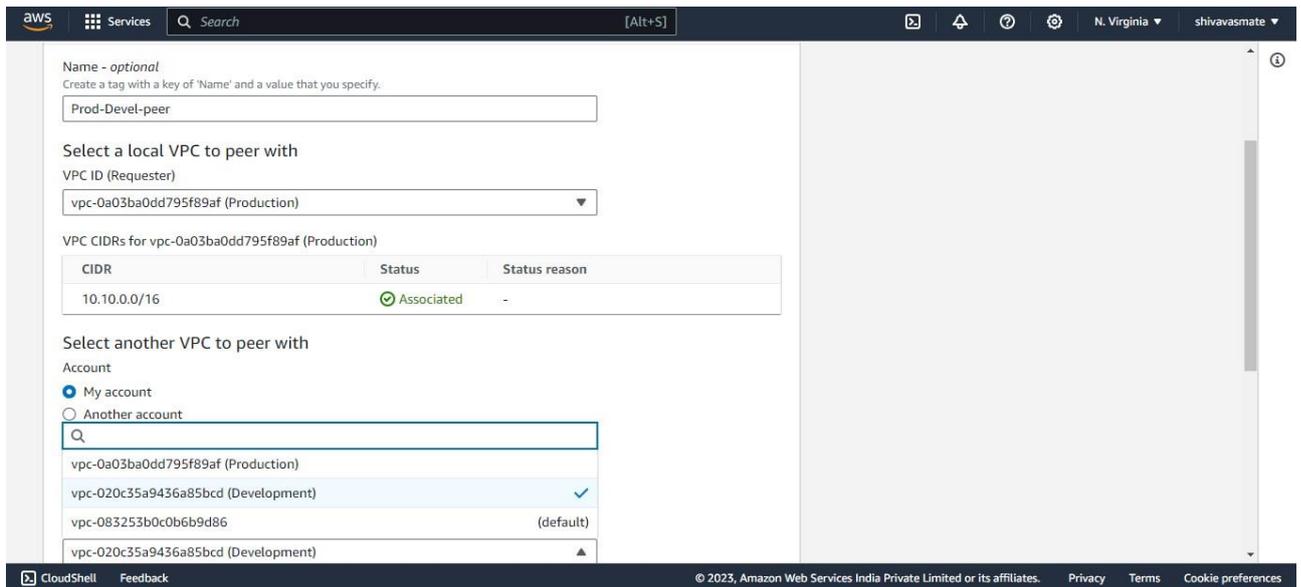
CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
```

PEERING CONNECTION

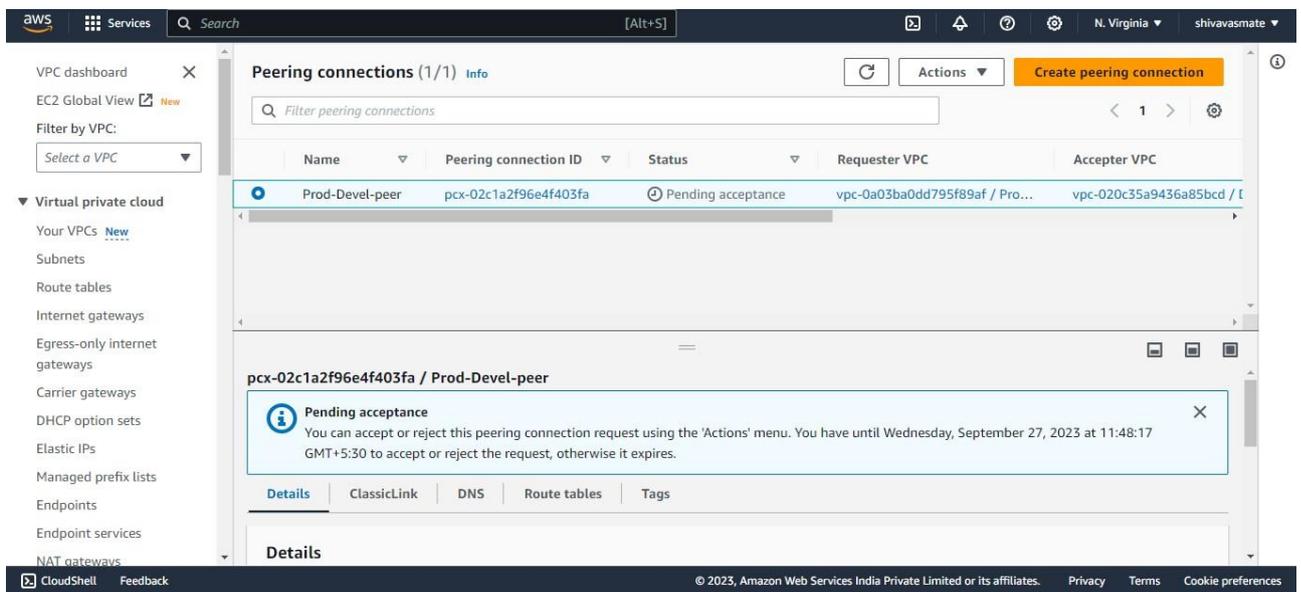
1. Go to peering connection in vpc and create peering connection



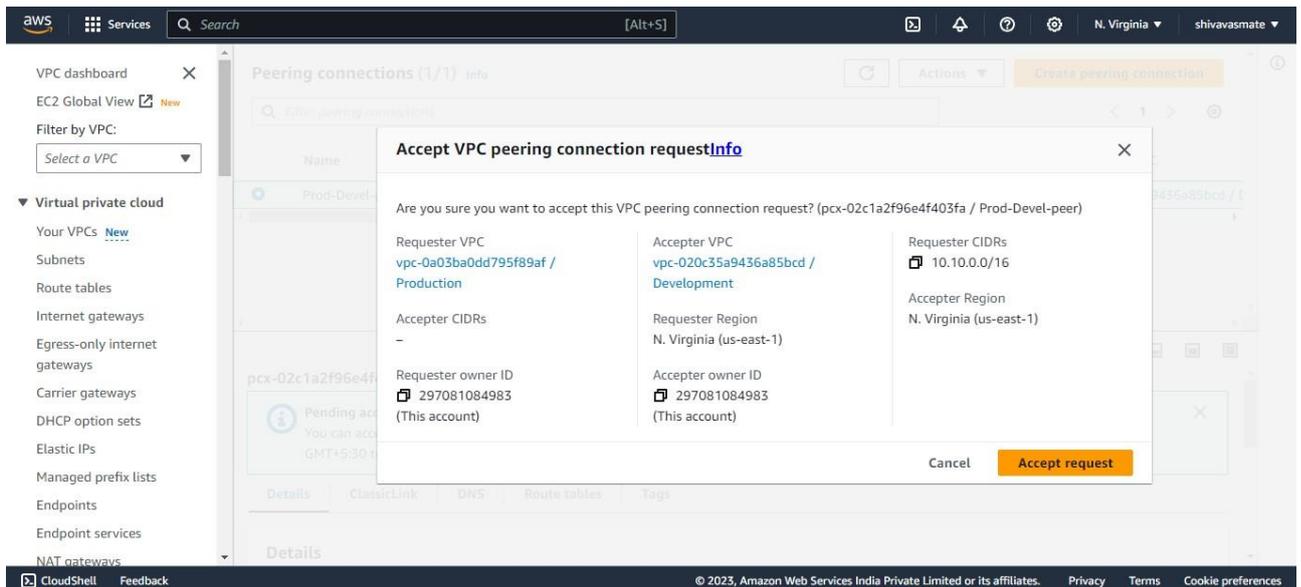
2. Now create peering connection with the name Prod-Devel-peer having requester vpc as production and acceptor vpc as development.



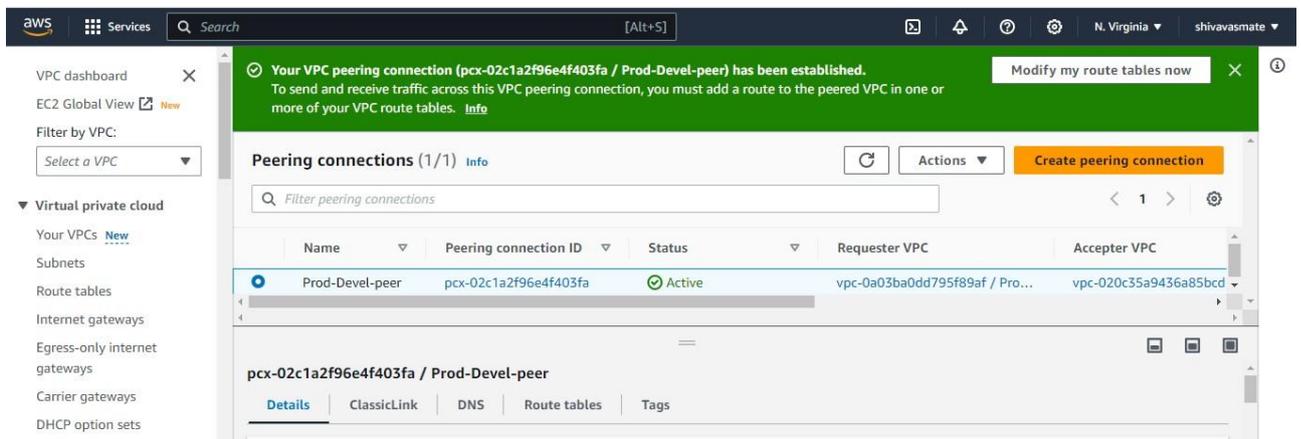
3. Peering connection is created with pending acceptance.



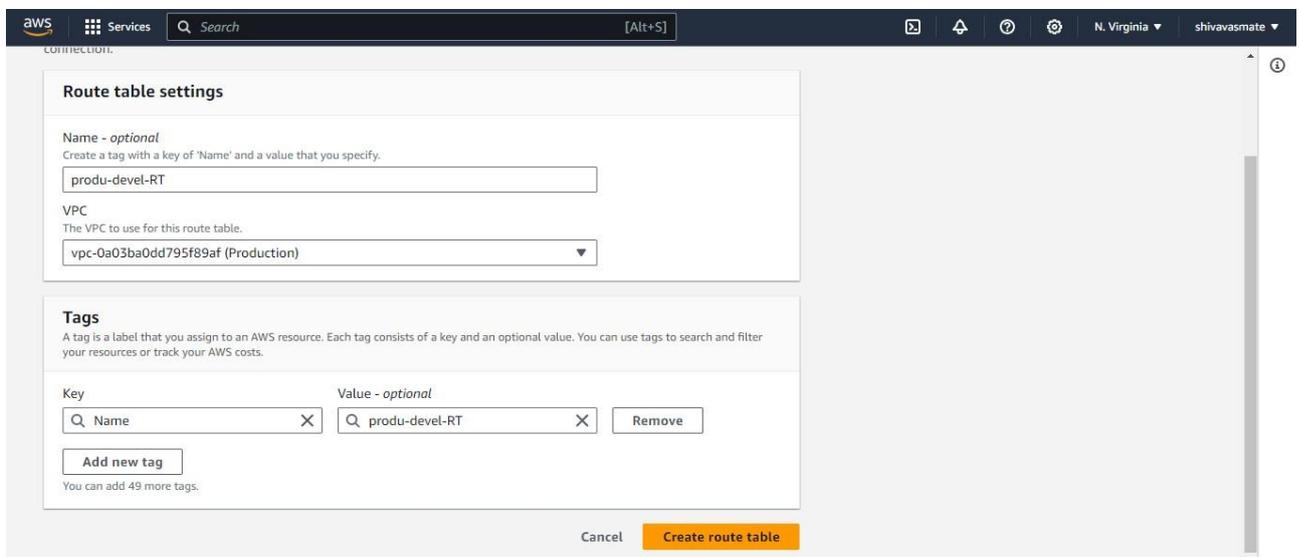
4. Go to action and accept request



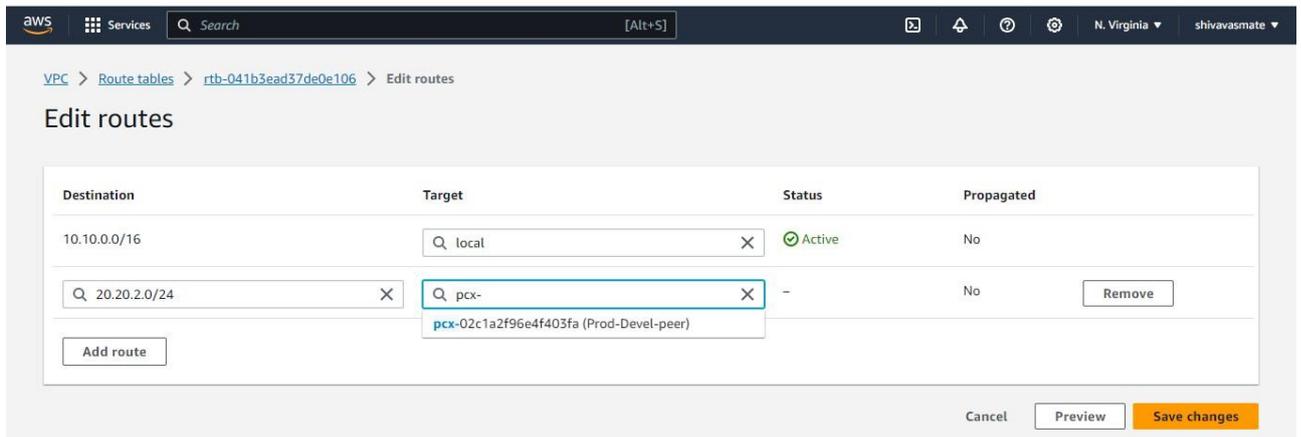
5. Now VPC peering connection is successful. It is now to modify route table.



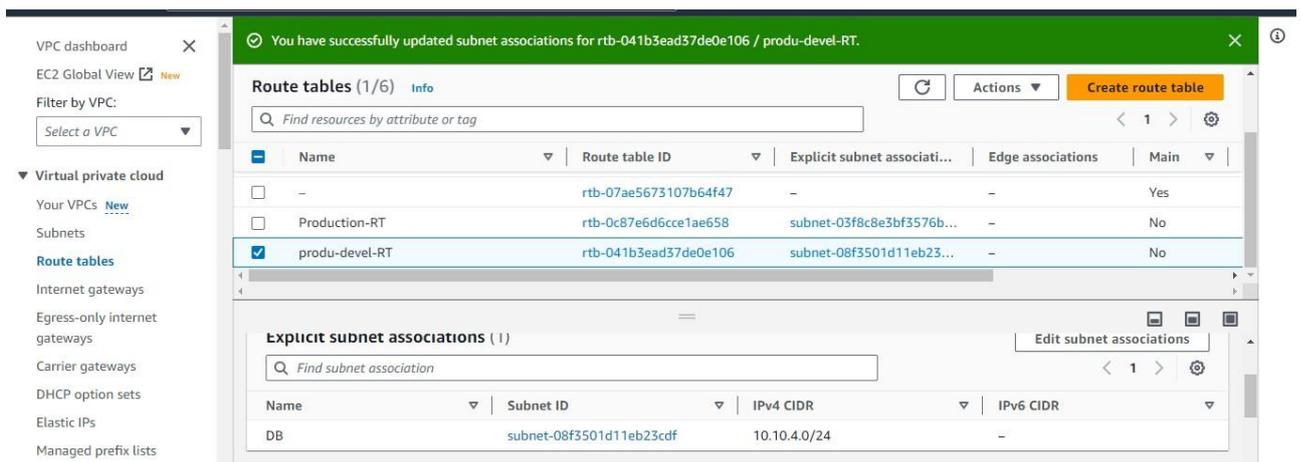
6. Go to route table and create route table with a name produ-devel-RT in the Production vpc.



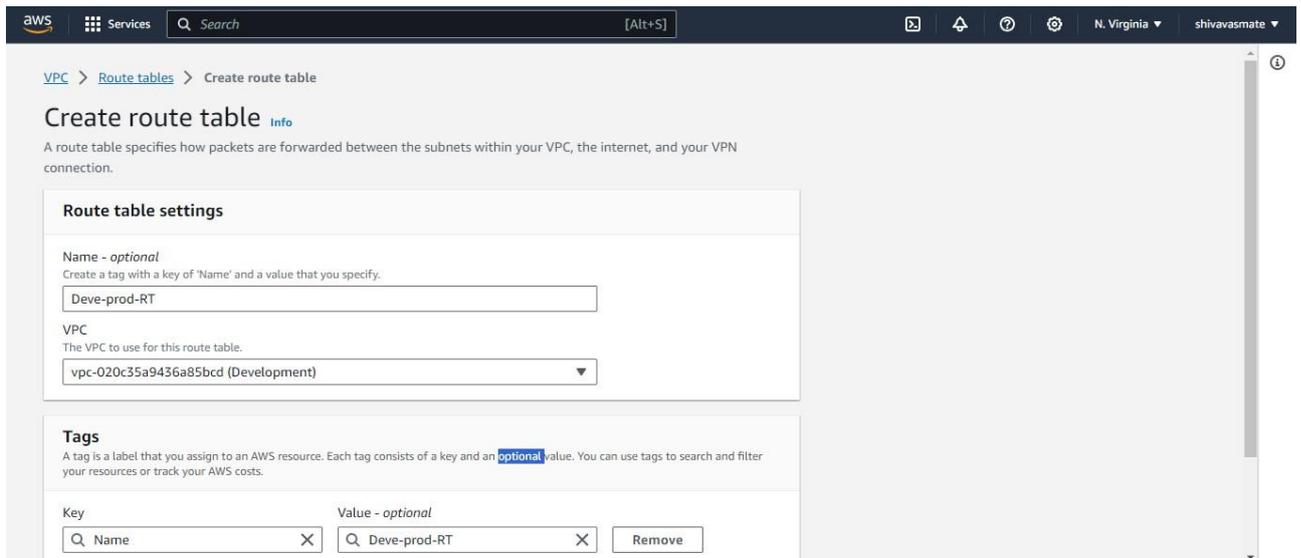
7. Edit route give the CIDR of the DB-2 (20.20.2.0/24) instance of the Development subnet



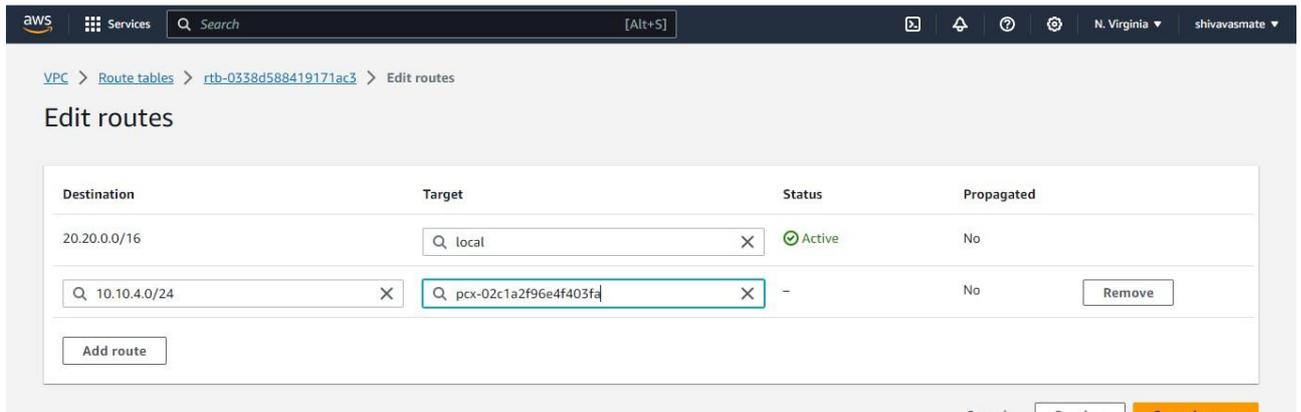
8. Make subnet association with DB (10.10.4.0/24) instance of the Production subnet.



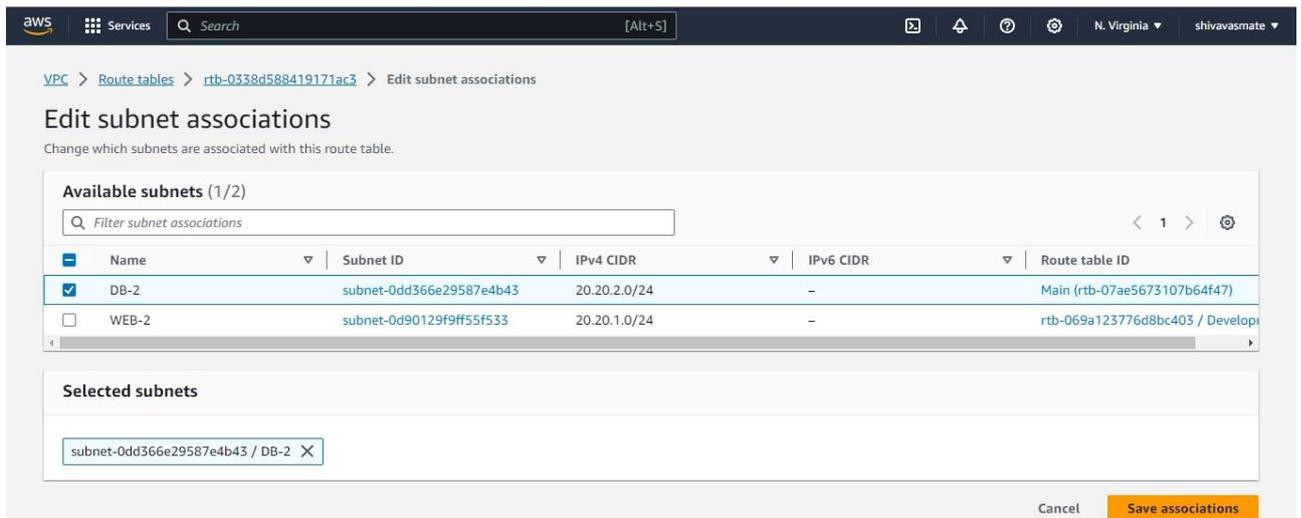
9. Similarly create another route table with a name Deve-prod-RT in the Development vpc.



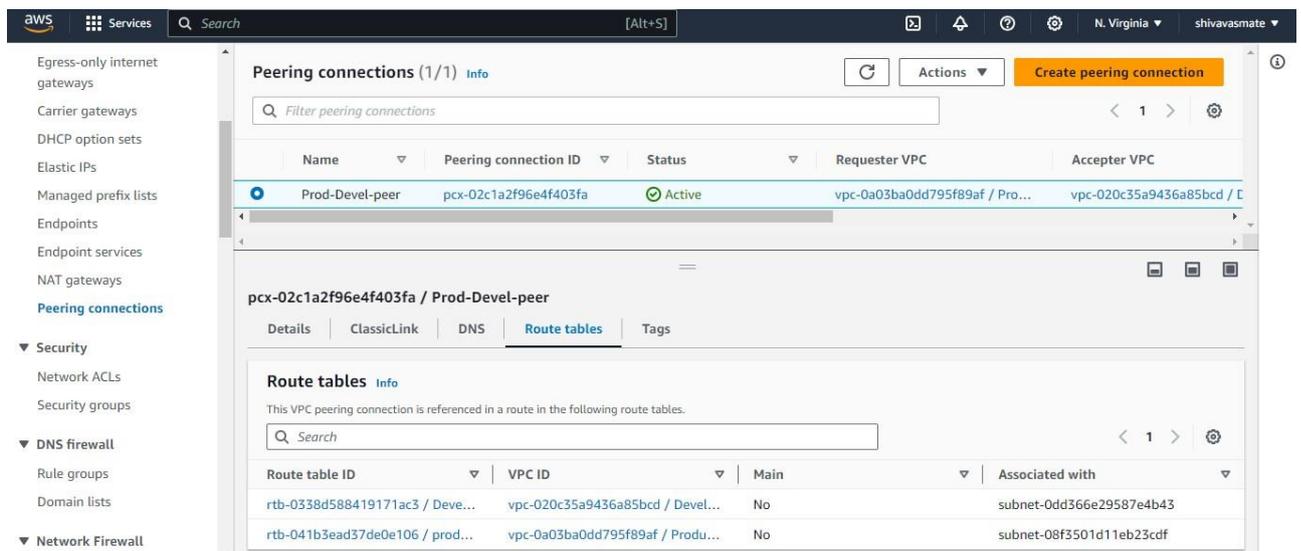
10. Edit route give the CIDR of the DB (10.10.4.0/24) instance of the Production subnet



11. Make subnet association with DB-2 (20.20.2.0/24) instance of the Development subnet.



12. Two-way peering connection is done



13. Now connect Web instance (10.10.5.230) ubuntu of production to DB instance(10.10.4.74)

linux by ssh -i test.pem ec2-user@10.10.4.74

