



## ASSIGNMENT -1

**COURSE:** DevOps

**Name:** D. Sai Rajitha

**MODULE:** Transit Gateway  
sairajithavarma@gmail.com

**Gmail:**

**TRAINER:** Mr. Madhukar sir

**Batch no:** 120 - 5pm

**Date:** 17-03-2024

---

### Creation of Transit Gateway in Two different accounts:

**Step1:** Create VPC in London region in an account. Now give name tag and CIDR value as show below

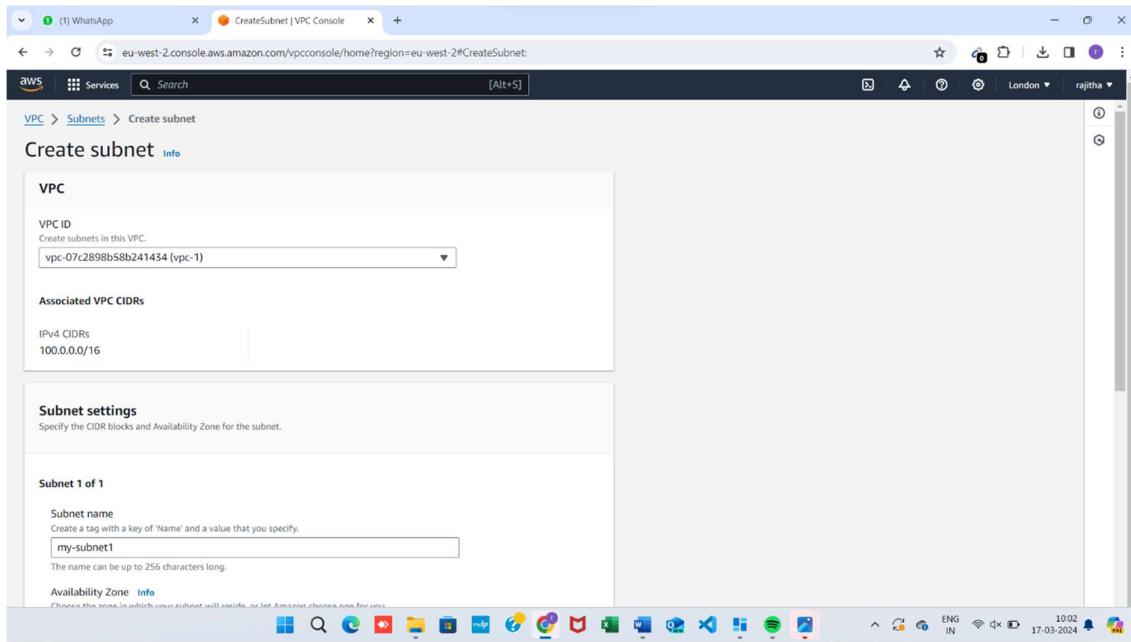
A screenshot of the AWS VPC Create VPC settings page. The page shows the following configuration:

- VPC settings:** "VPC only" is selected. "Name tag - optional" is set to "vpc-1".
- IPv4 CIDR:** "100.0.0.0/16" is entered.
- IPv6 CIDR block:** "No IPv6 CIDR block" is selected.

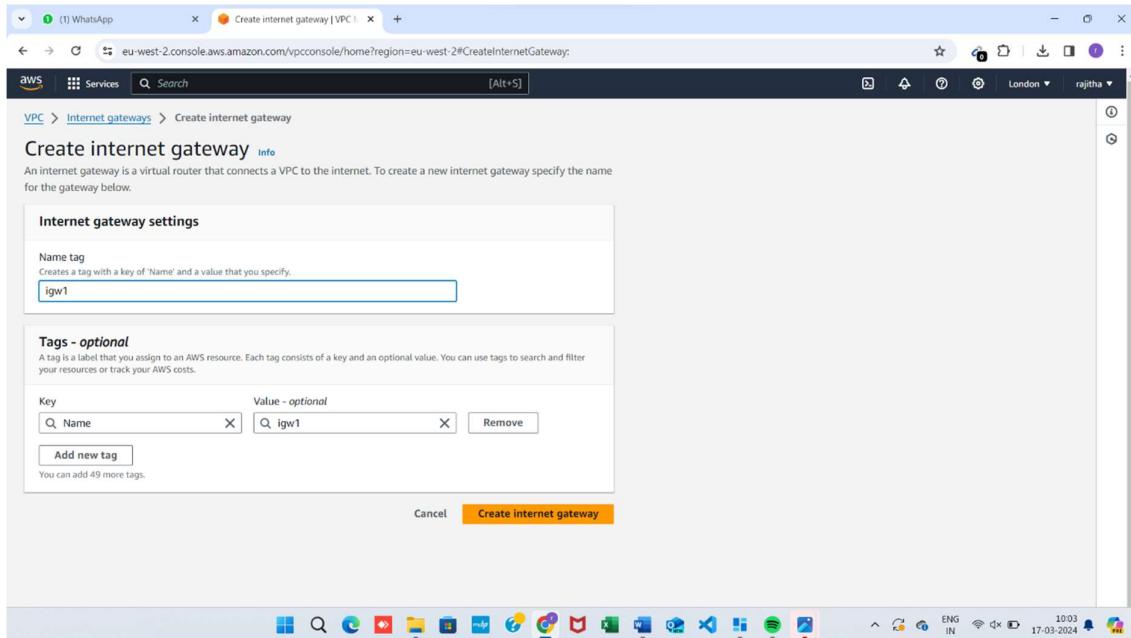
The browser title is "CreateVpc | VPC Console" and the URL is "eu-west-2.console.aws.amazon.com/vpcconsole/home?region=eu-west-2#CreateVpc:createMode=vpcOnly".

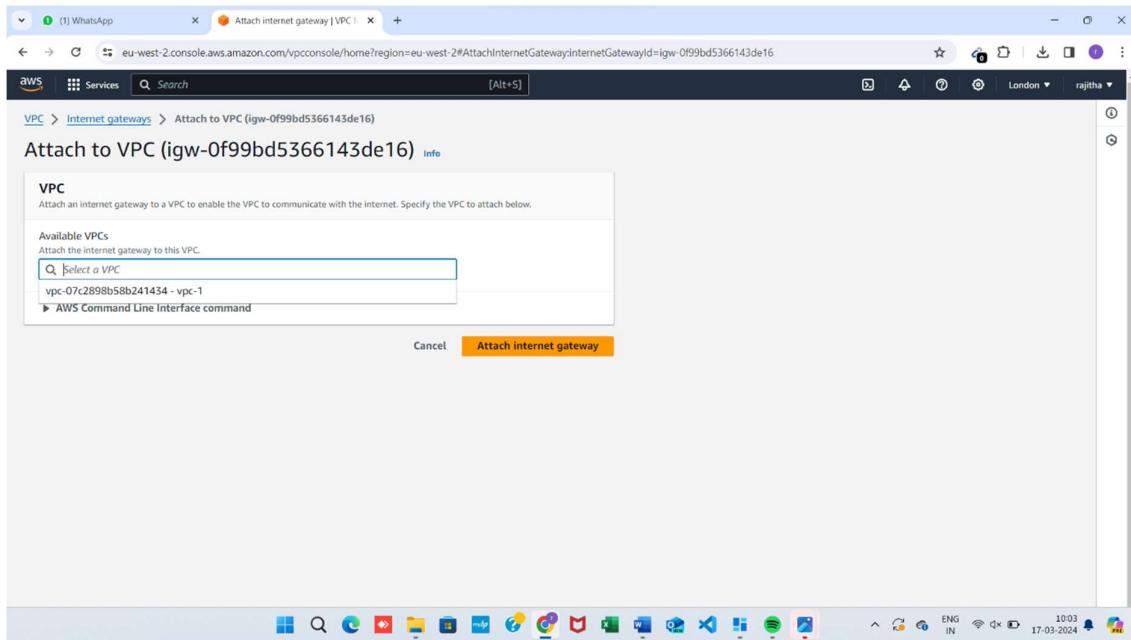
**Step2:** Create a subnet

Select VPC id name your subnet ,Select availability zone and give some CIDR value create subnet

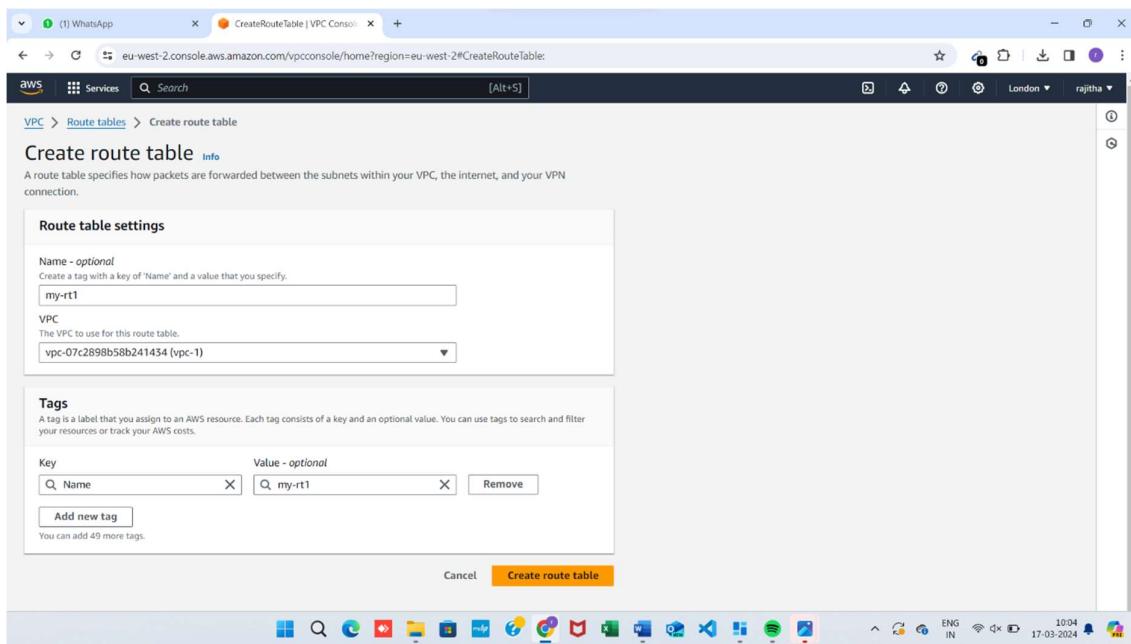


**Step3:** Now create IGW(internet gateway) attach it VPC as show below





#### Step4: create routetables



**Step5:** Now edit route attach Igw to routetable means now we are giving public access and also edit the subnet associations and attach to the the subnet

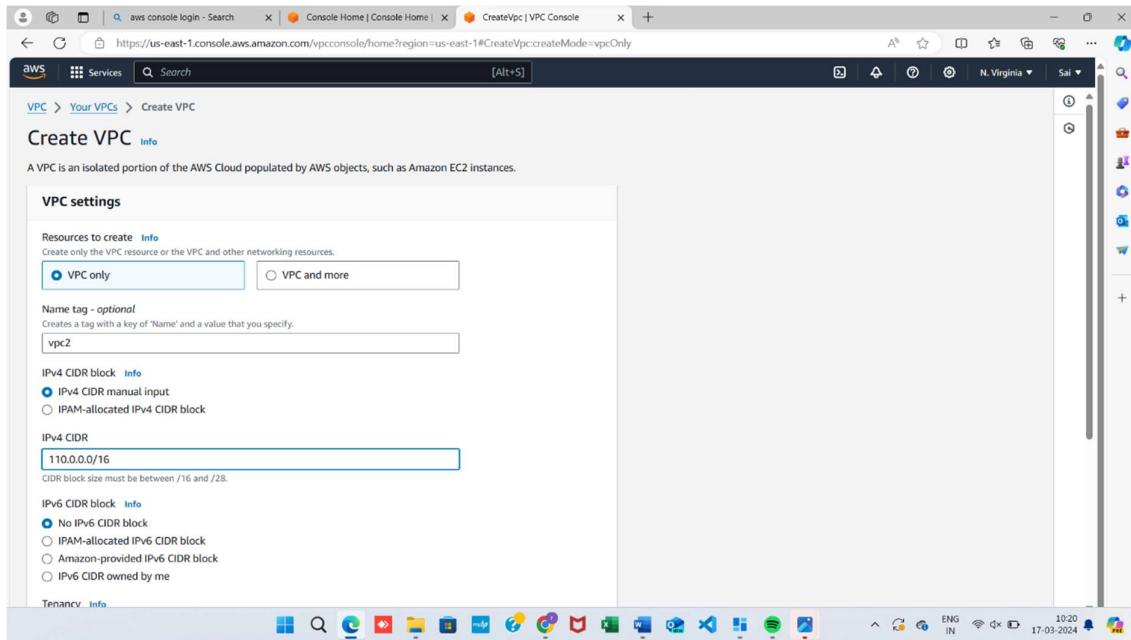
The screenshot shows the AWS VPC Console interface. A modal window is open confirming the creation of a route table named 'rtb-02d9a7d662371adbc / my-rt1'. The main pane displays the details of this route table, including its ID, association with a VPC ('vpc-07c2898b58b241434 | vpc-1'), and a single route entry for '100.0.0/16' pointing to 'local' with an active status. An 'Actions' dropdown menu is visible on the right, listing options like 'Set main route table', 'Edit subnet associations', and 'Edit routes'. The left sidebar shows navigation links for VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and other VPC-related services.

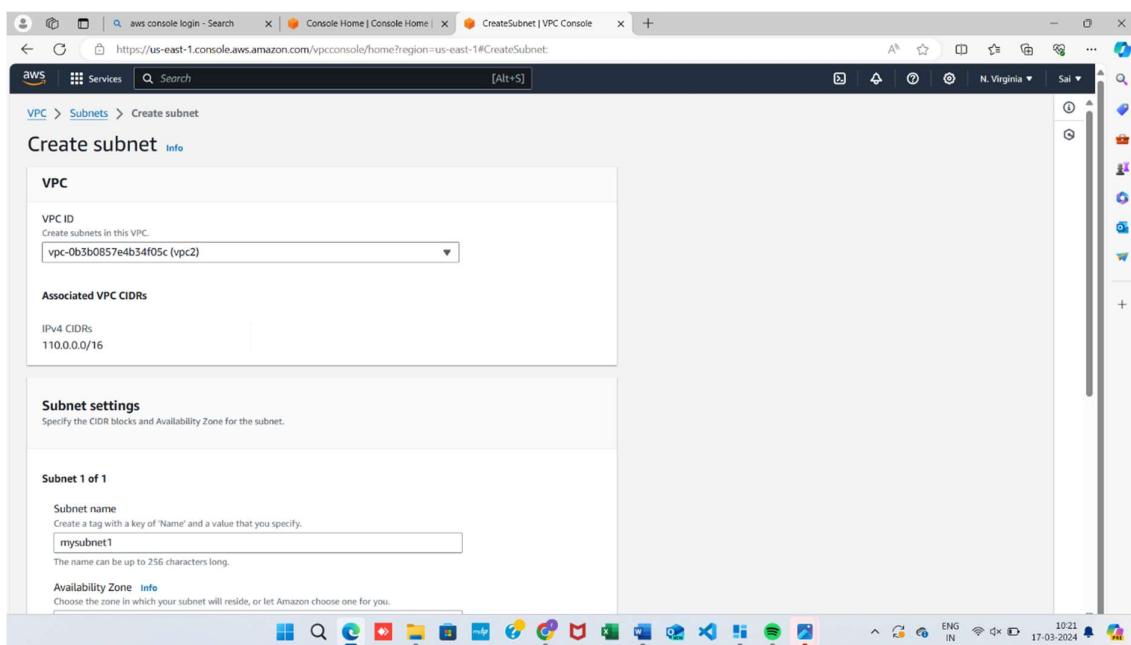
This screenshot shows the 'Edit routes' page for the route table 'rtb-02d9a7d662371adbc'. It lists two routes: one for '100.0.0/16' targeting 'local' (status: Active, propagated: No) and another for '0.0.0/0' targeting an Internet Gateway (status: In Progress, propagated: No). The 'Save changes' button is highlighted at the bottom right. The interface includes a search bar and a toolbar with various icons.

**Step6:** So now login to another account select another region as per your convenience repeat the process what we have done previously

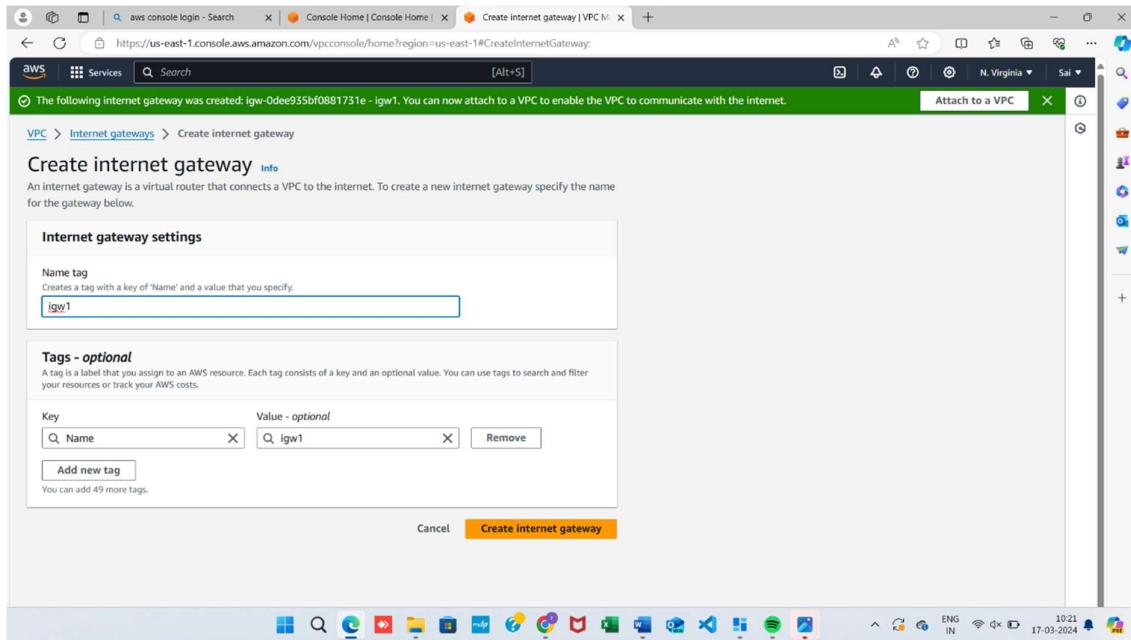
**Step7:** create VPC



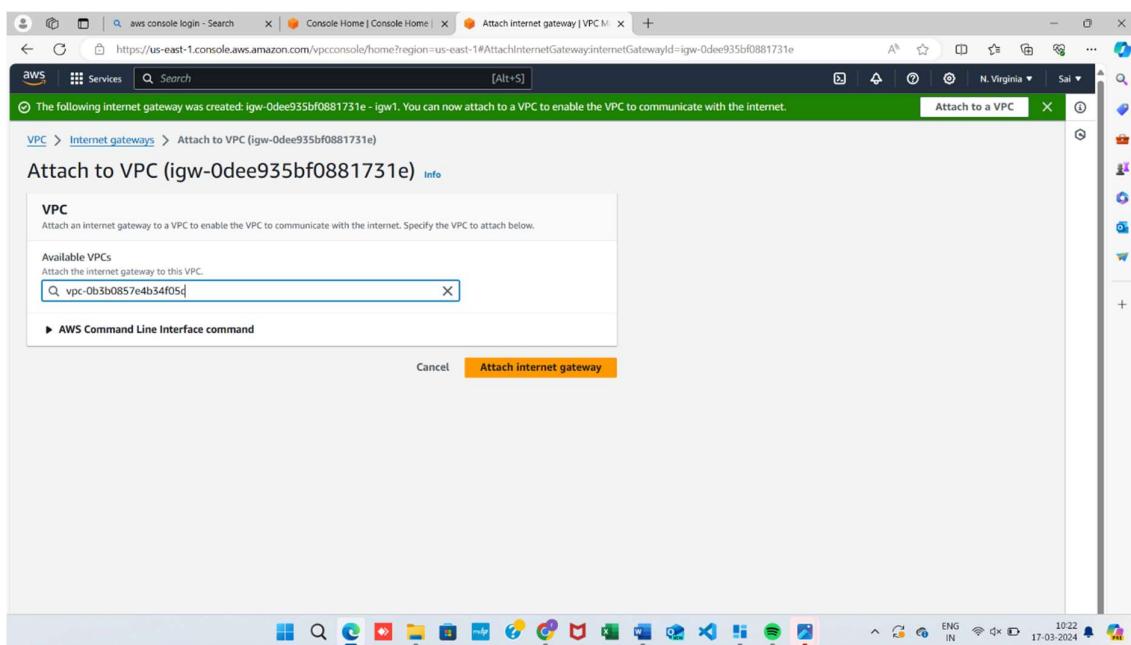
### Step8: create subnet



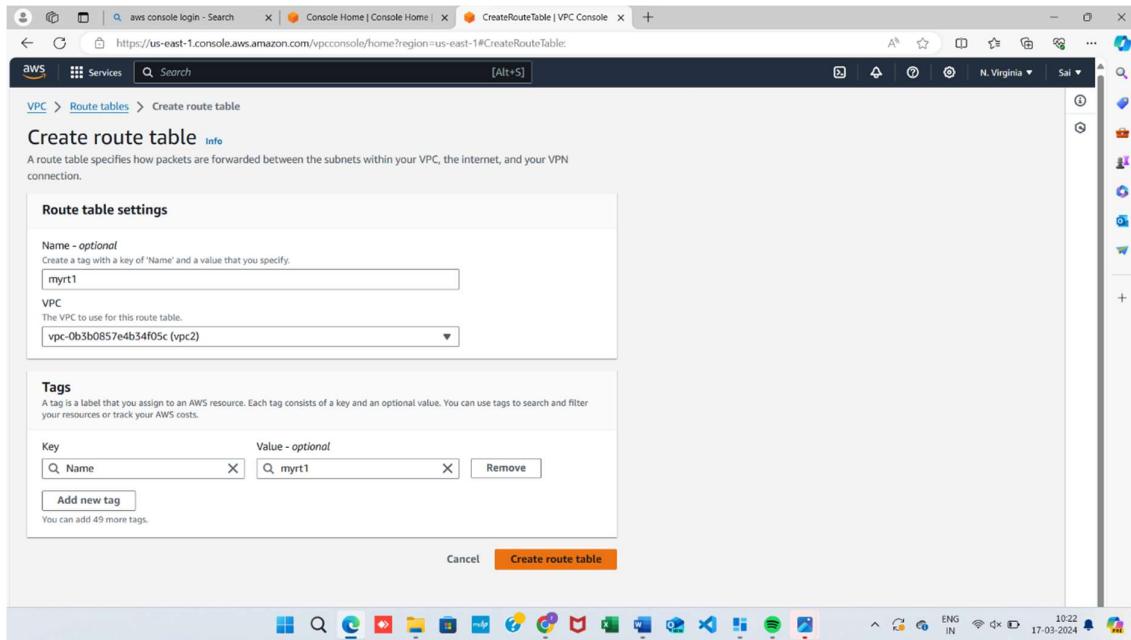
### Step9: Create igw



### Step10: attach it to VPC



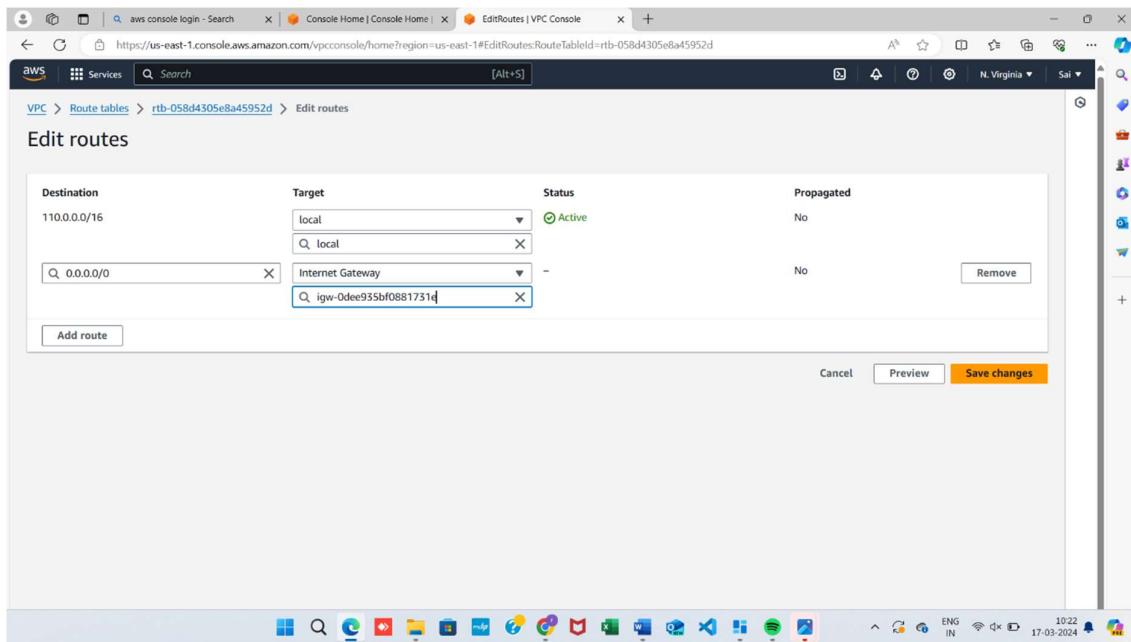
### Step11: Now create route tables



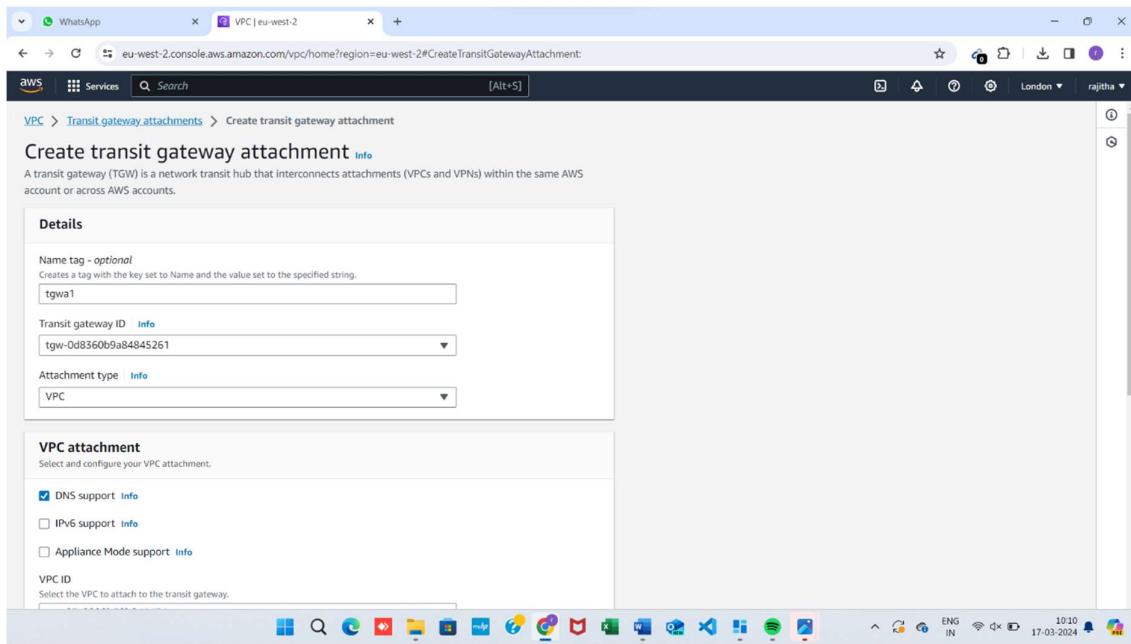
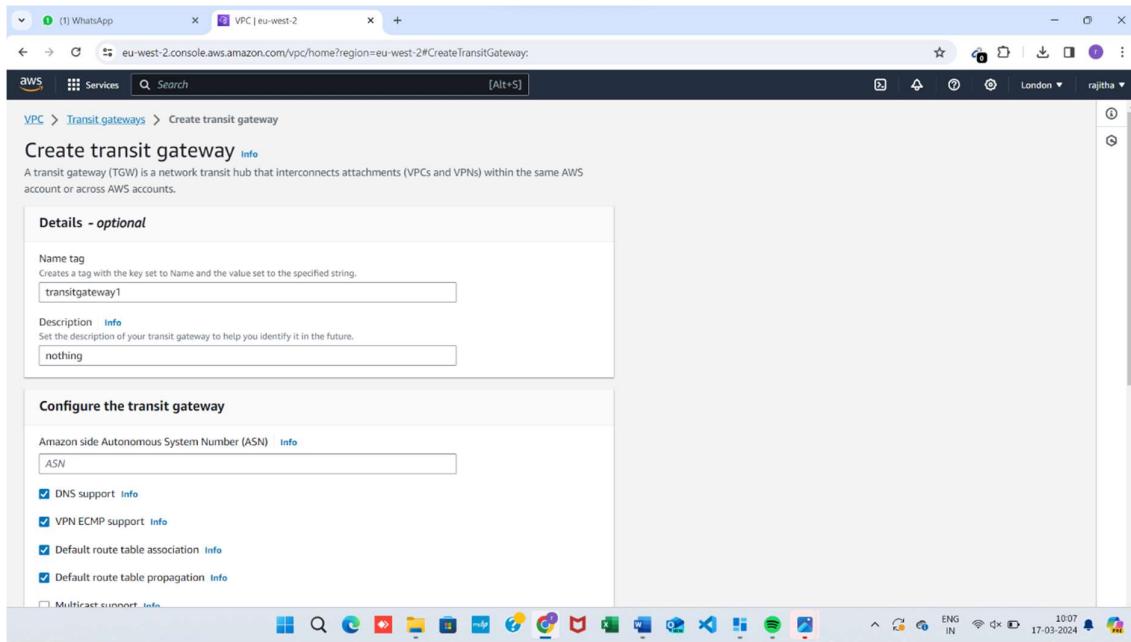
### Step12: Edit route

Add route attach internet gateway to route table

And edit subnet associations and attach to the subnet



**Step13:** Now go to the previous account now create transit gateway and then create transit gateway attachment



**Step14:** Create the transit gateway and attachment in another account which is in virginia

aws console login - Search | Console Home | VPC | us-east-1

VPC > Transit gateways > Create transit gateway

### Create transit gateway Info

A transit gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same AWS account or across AWS accounts.

**Details - optional**

Name tag Info  
Creates a tag with the key set to Name and the value set to the specified string.

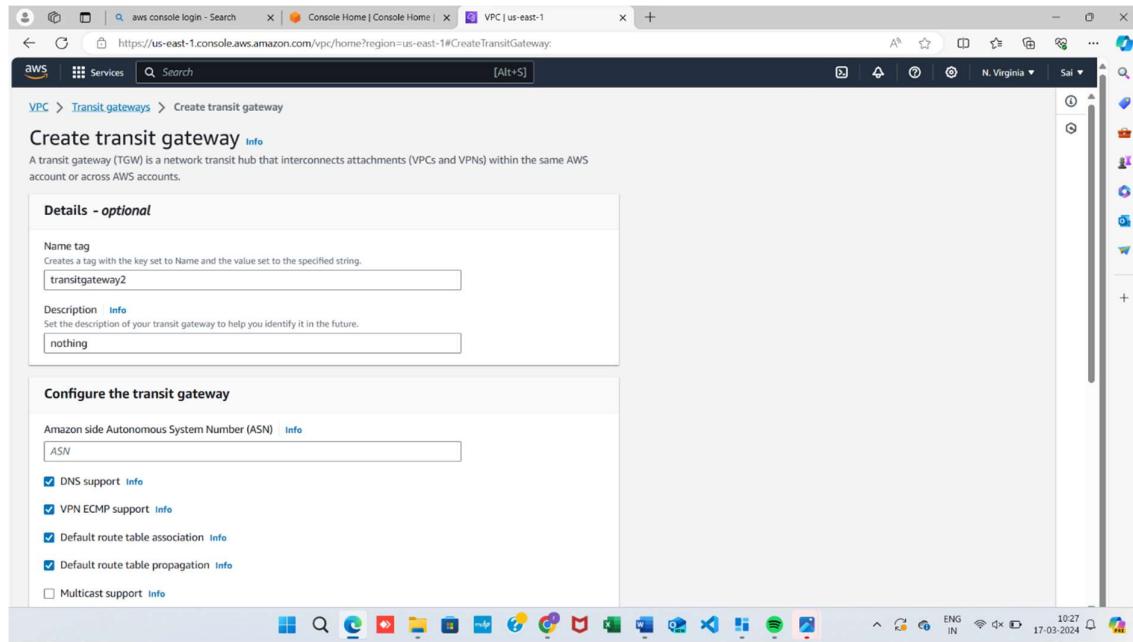
Description Info  
Set the description of your transit gateway to help you identify it in the future.

**Configure the transit gateway**

Amazon side Autonomous System Number (ASN) Info

DNS support Info  
 VPN ECMP support Info  
 Default route table association Info  
 Default route table propagation Info  
 Multicast support Info

10:27 17-03-2024



aws console login - Search | Console Home | VPC | us-east-1

VPC > Transit gateway attachments > Create transit gateway attachment

### Create transit gateway attachment Info

A transit gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same AWS account or across AWS accounts.

**Details**

Name tag - optional Info  
Creates a tag with the key set to Name and the value set to the specified string.

Transit gateway ID Info

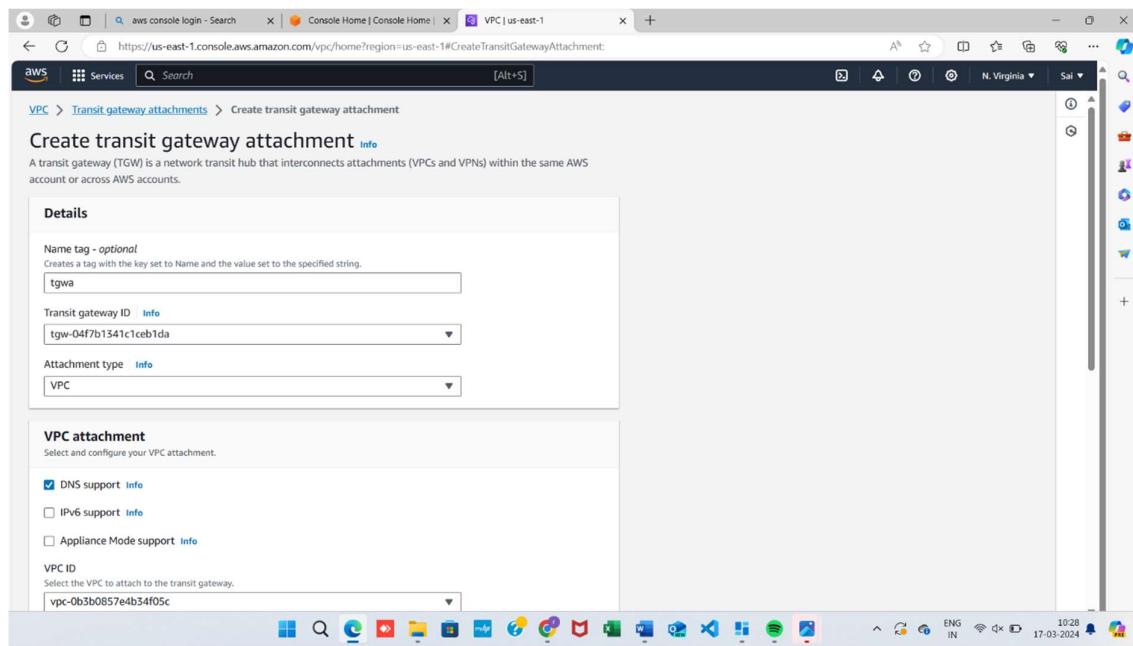
Attachment type Info

**VPC attachment**  
Select and configure your VPC attachment.

DNS support Info  
 IPv6 support Info  
 Appliance Mode support Info

VPC ID  
Select the VPC to attach to the transit gateway.

10:28 17-03-2024



**Step15:** Now attach transit gateway to the route table in the place of IP give the IP address of other VPC which was created in another account i.e in London as show in the above picture

**Step16:** Do the same with the previous account which is in connect transit gateway to the route table repeat the process same as Step15

**Step17:** Now again create transit gateway attachment now in the place attachment type we have to select the Peering Connection select other account copy the parallel account number ,Select the region then create the transit attachment

WhatsApp VPC | eu-west-2 eu-west-2.console.aws.amazon.com/vpc/home?region=eu-west-2#CreateTransitGatewayAttachment:

**Peering connection attachment**

Select and configure your peering connection attachment.

**Account** **Info**  
 My account  
 Other account

**Account ID**  
Add 12 digits AWS account number (without hyphens) of the transit gateway you are peering with.

**Region** **Info**  
US East (N. Virginia) (us-east-1)

**Transit gateway (accepter)** **Info**

**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	Value - optional
<input type="text" value="Name"/>	<input type="text" value="tgwa123"/>

**Add new tag**  
You can add up to 49 more tags.

**Create transit gateway attachment**

aws console login - Search Console Home | Console Home Transit gateway attachments | VI

https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#TransitGatewayAttachments:

**Transit gateway attachments (2) info**

**Create transit gateway attachment**

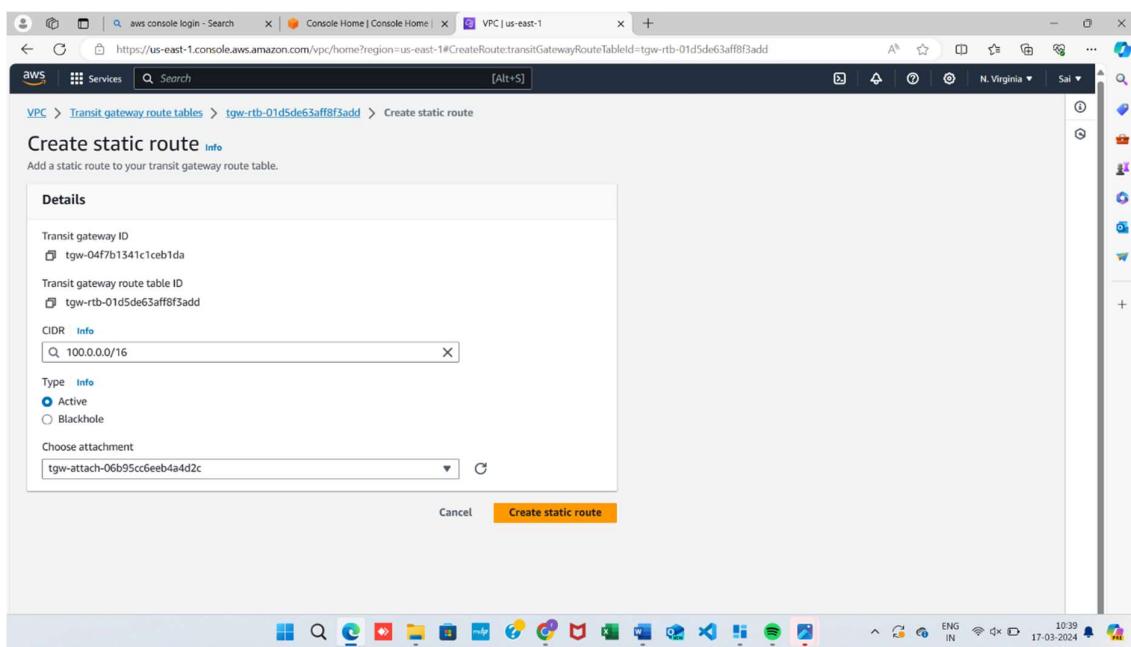
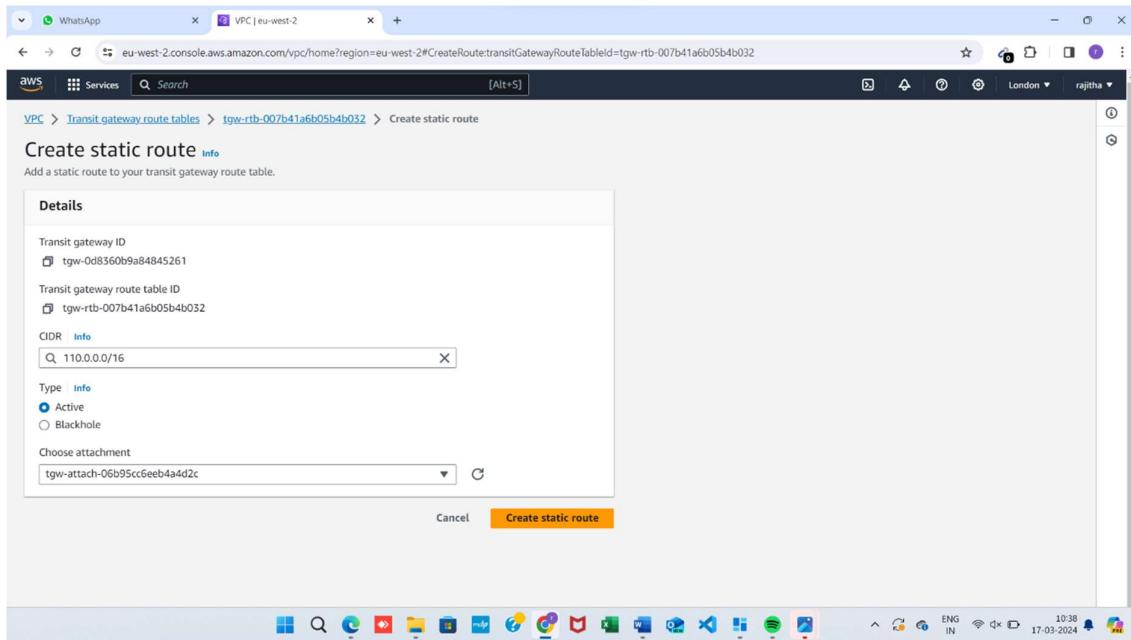
Name	Transit gateway attachment ID	Transit gateway ID	State	Resource t...	Resource ID
tgw-attach-06b95cc6eeb4a4d2c	tgw-04f7b1341c1ceb1da	Pending Acceptance	Peering	tgw-0d8360b9a1	
tgwa	tgw-attach-0d0864a632aea5824	tgw-04f7b1341c1ceb1da	Available	VPC	vpc-0b3b0857e4

**Select a transit gateway attachment**

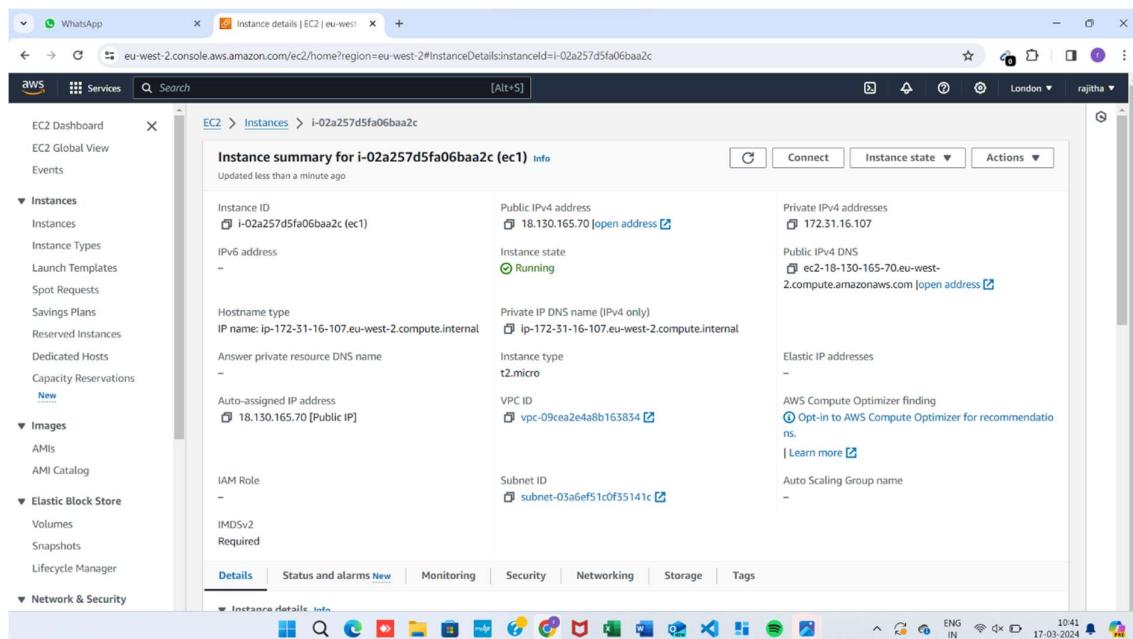
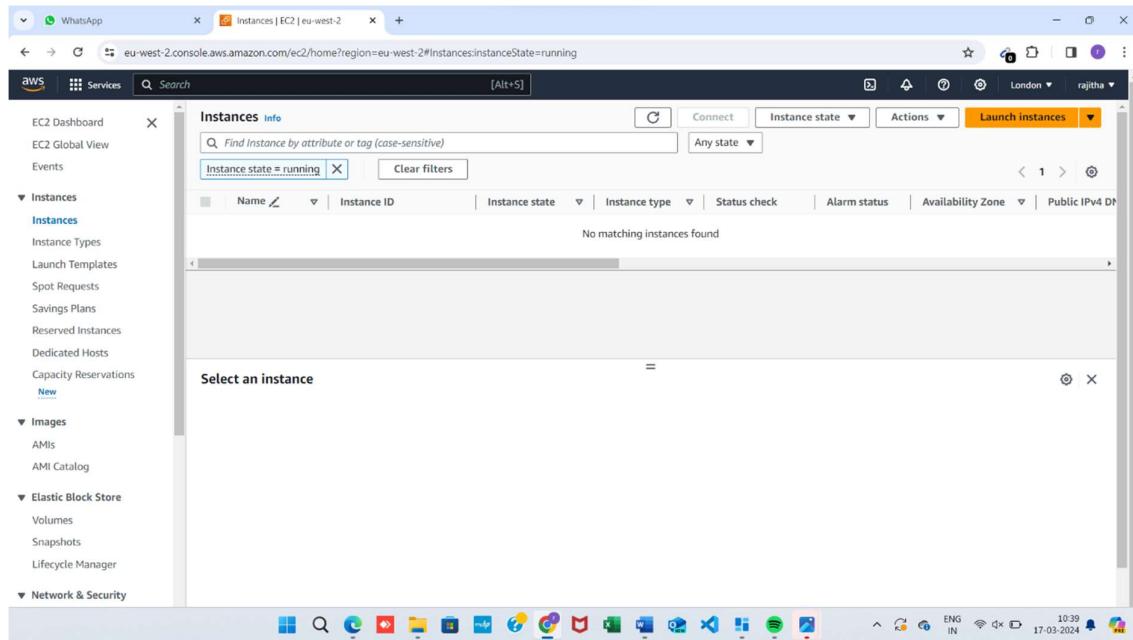
**Step18:** As you can see in the above picture we get a pending acceptance

**Step19:** click on that go to actions accept the peering attachment

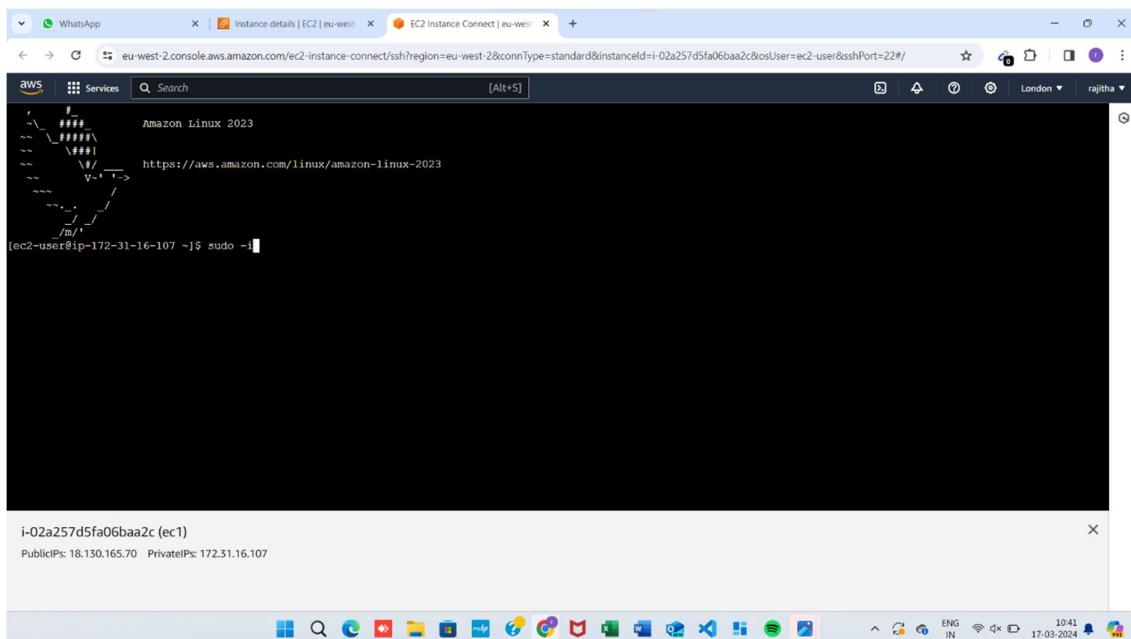
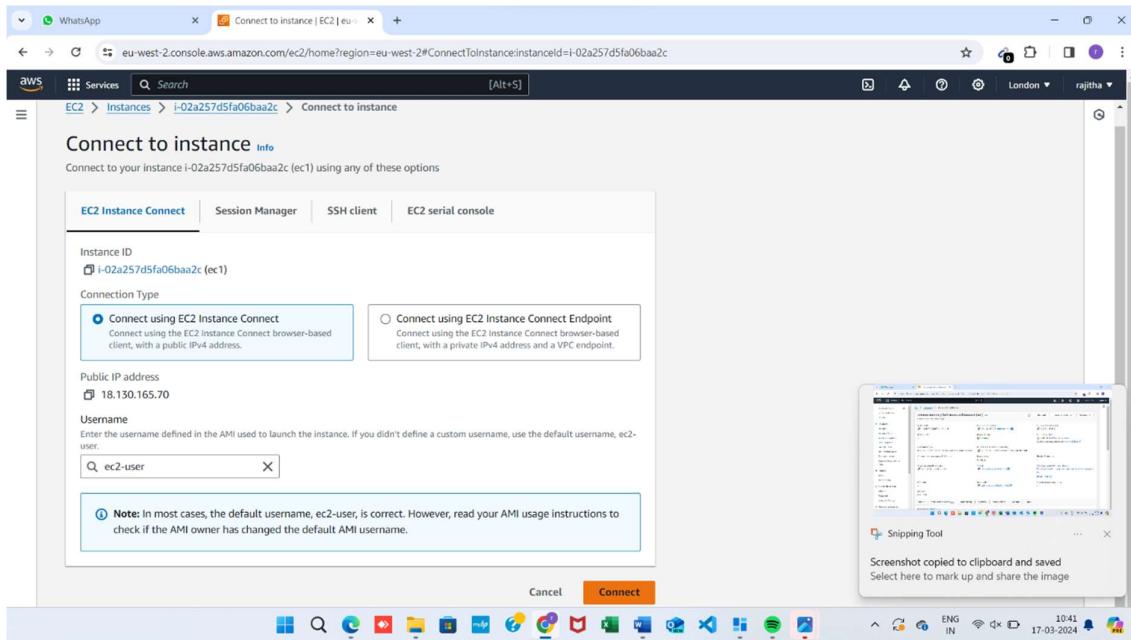
Both the servers get connected so these transit gateway attachments you should attach it to transit route tables create static route table



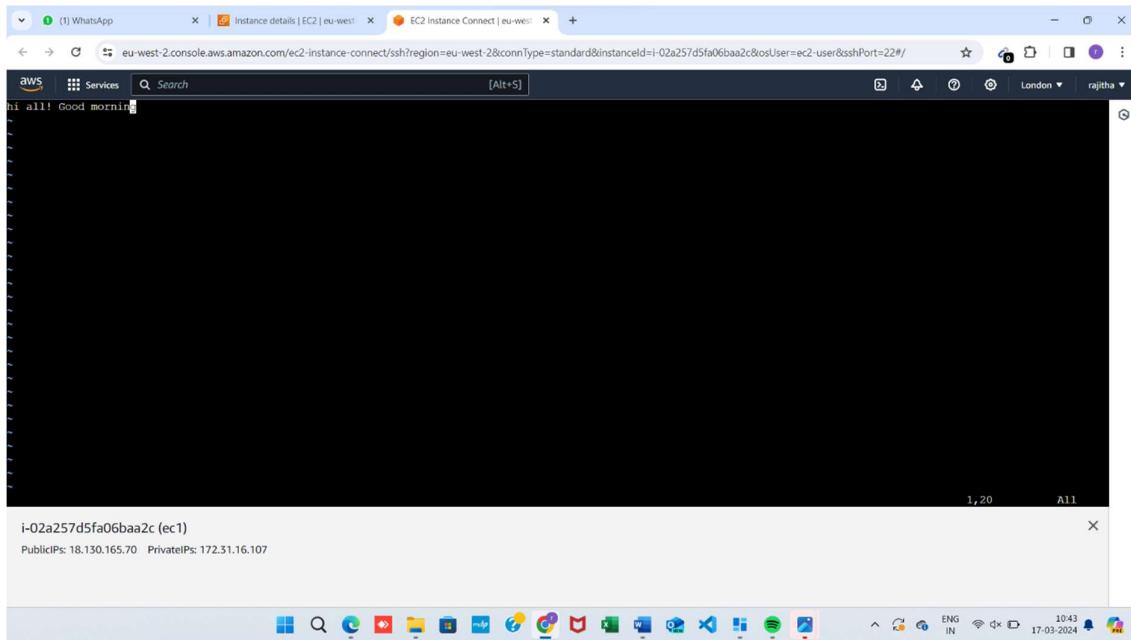
**Step20:** Now Create Ec2 instance tag, Ami tag, Instance type, keypair ang edit networking setting select vpc subnet enable and create instance as show in the below



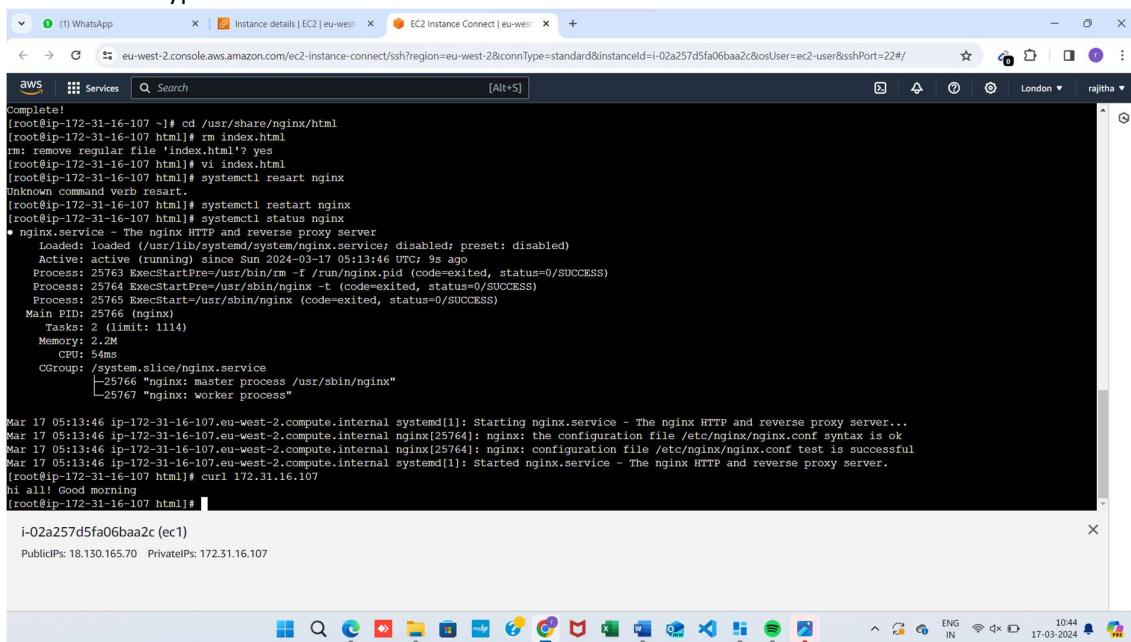
## Step21: Connect to the instance



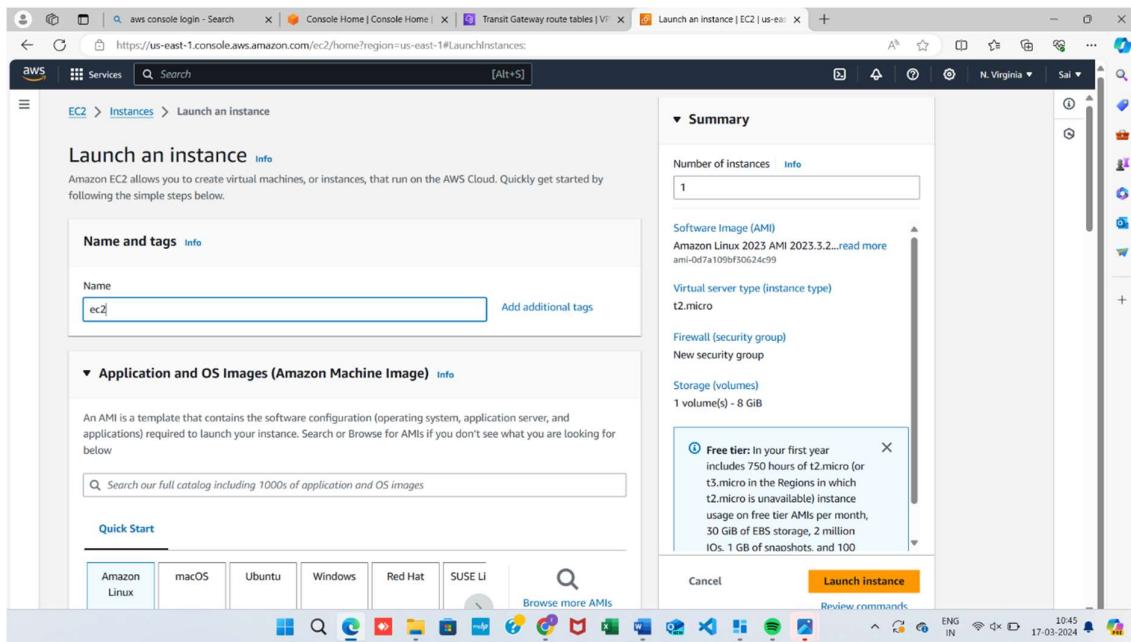
**Step22:** “sudo -i”, “yum update -y”, “yum install nginx -y”, “cd /usr/share/nginx/html”, “rm nginx.html”, “vi index.html”, insert data,



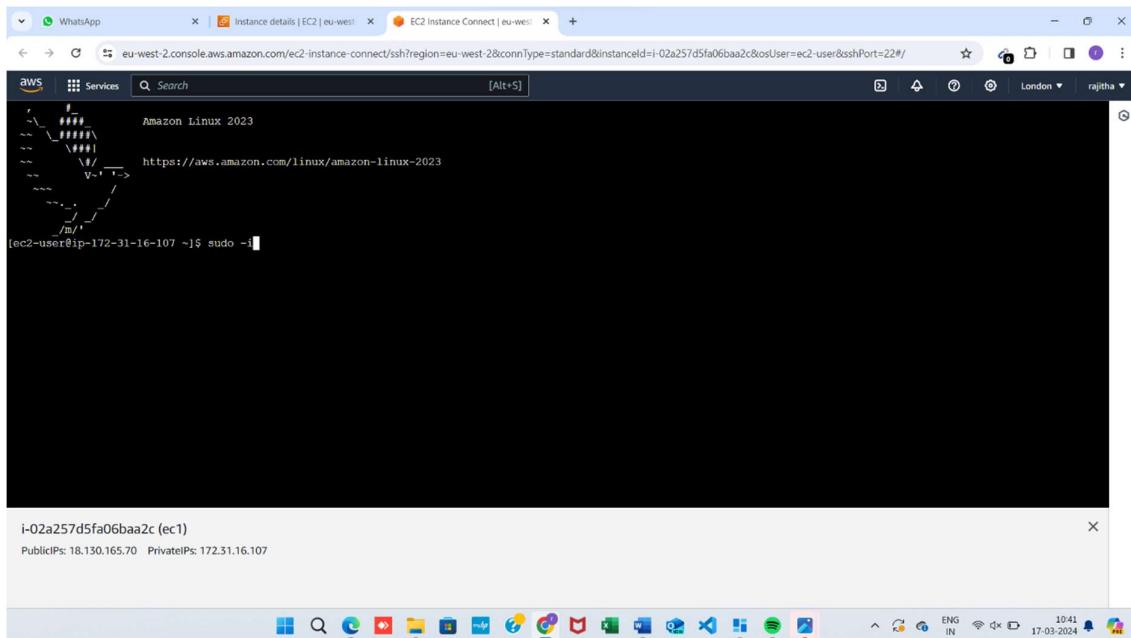
**Step23:** systemctl restart nginx, systemctl status nginx, now curl paste the public ip address now can see the which you have inserted in the file



**Step24:** Now install the Ec2 instance in other account which is in virginia



Step25: Repeat Step 22 and 23



The screenshot shows a CloudShell terminal window with the following content:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-100-0-0-67 ~]$ sudo -i
[root@ip-100-0-0-67 ~]# yum update -y
Last metadata expiration check: 0:01:29 ago on Tue Mar 26 16:18:24 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-100-0-0-67 ~]# curl 172.31.36.162
hi all! Good morning
[root@ip-100-0-0-67 ~]#
```

Below the terminal output, the instance details are shown:

i-010a7cfa58457d06c (ec2)  
Public IPs: 18.171.56.138 Private IPs: 100.0.0.67

**Step26:** You can see the data which you have created in account 1 can be showed in account 2

So I conclude this is the use of transit gateway we can see the data or connect to the different isolated servers which are in different regions or different accounts