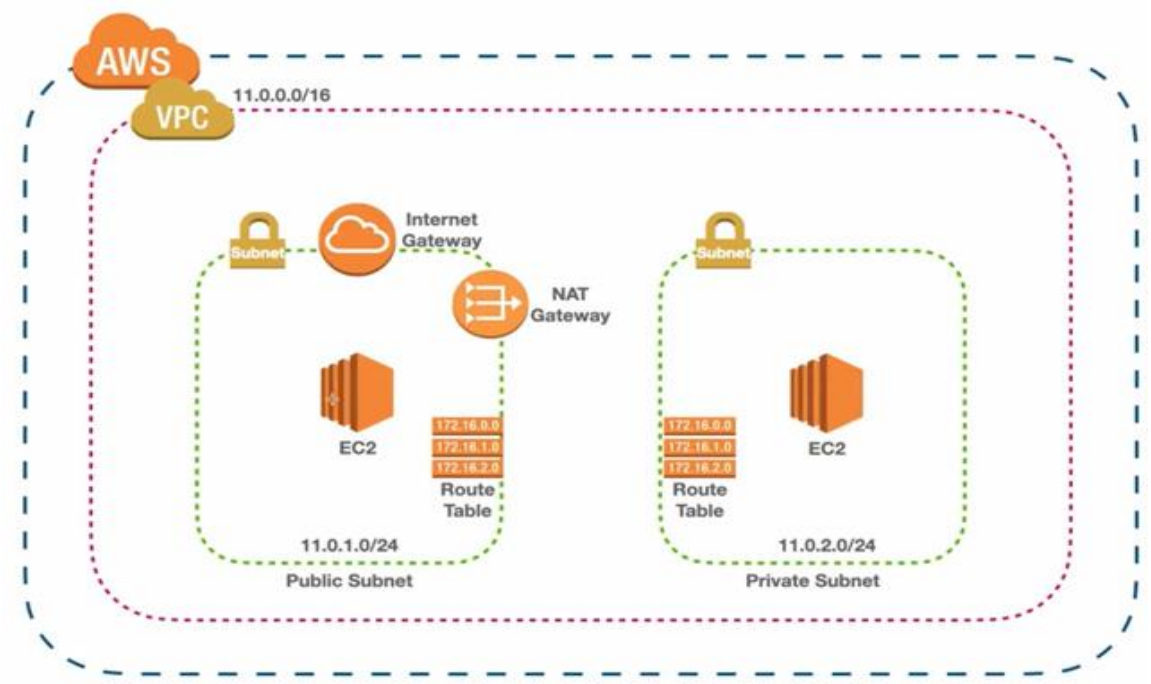
HOW TO SETUP EC2 INSTANCE IN VPC

AWS VPC (Amazon Virtual Private Cloud): It is a service that allows you to create a private network in the cloud. It gives you full access over your networking environment allowing you to define your IP address range also to create subnets and manage routing tables. With VPC you can securely connect your resources to the internet or keep them isolated from external traffic depending on your needs. It's like building your own private data center within AWS.



Internet Gateway: An Internet Gateway works by establishing a connection between a VPC and the internet. The VPC must have a public subnet, and the instances within that subnet must have a public IP address to communicate with the internet. An Internet Gateway acts as a bridge between the VPC and the internet. An Internet Gateway is commonly used when you want resources within a VPC to be accessible from the internet.

NAT Gateway: A NAT Gateway enables instances in a private subnet to connect to the internet or other AWS services but prevents the internet or other AWS services from initiating a connection with those instances. A NAT Gateway is commonly used when you have resources within a private subnet that require outbound internet access but should not be directly accessible from the internet.

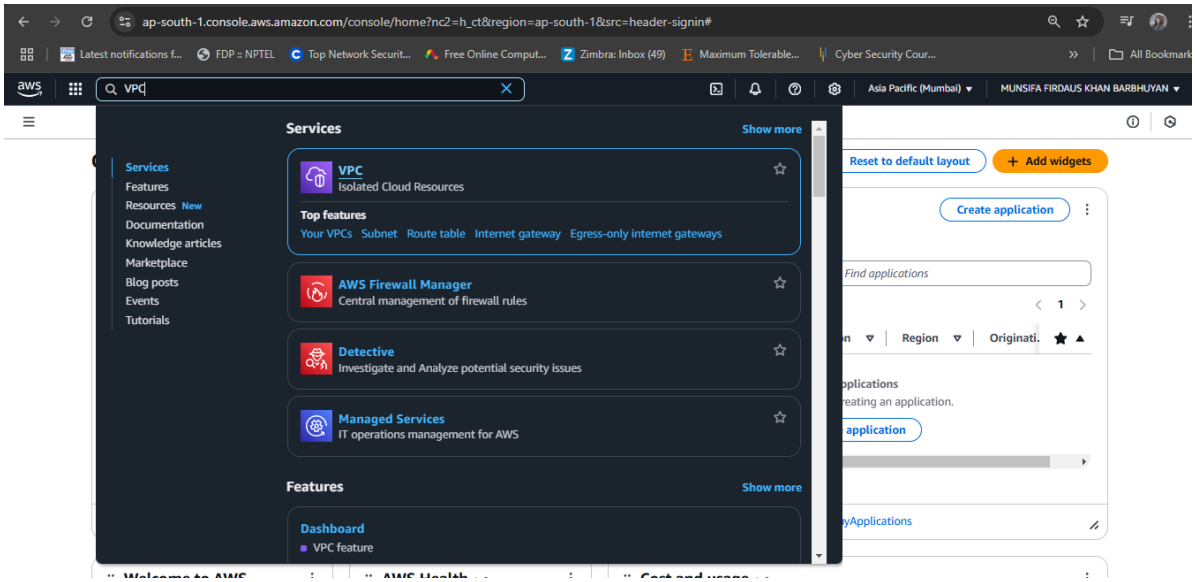
Subnet: A subnet is like a smaller group within a large network. It is a way to split a large network into smaller networks so that devices present in one network can transmit data more easily.

- **The public subnet has instances with public IP addresses allowing them to communicate directly with the internet. These instances can be web servers load balancers or other publicly accessible services. An Internet Gateway connects the public subnet to the internet providing inbound and outbound communication.**

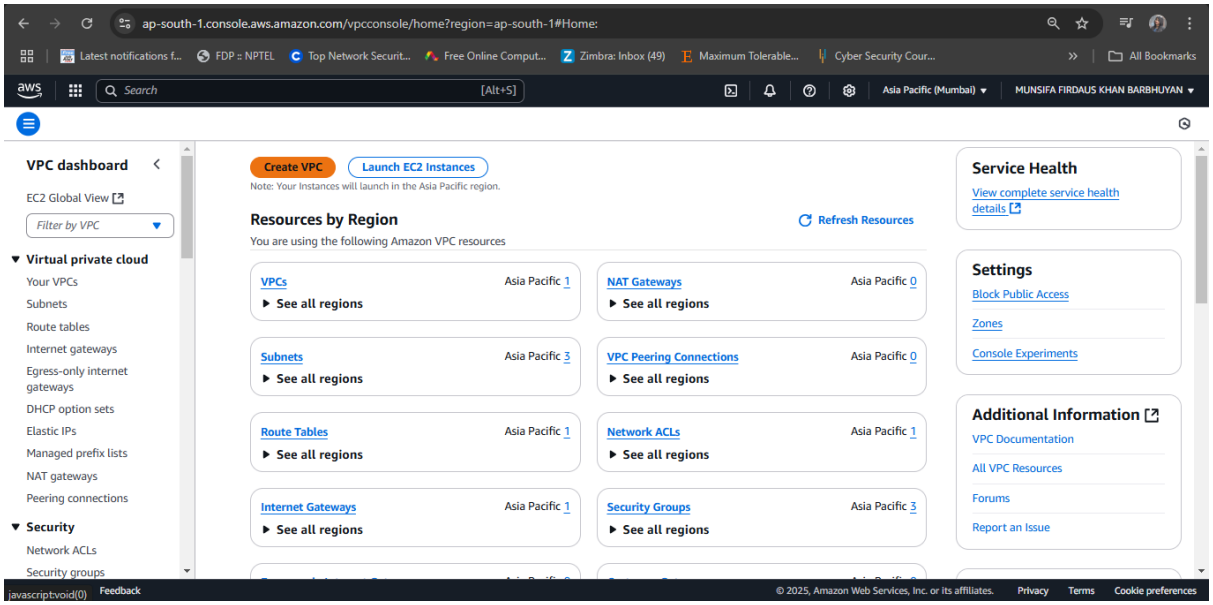
- **The private subnet has instances with private IP addresses. These instances cannot directly access the internet but can connect to resources within the same VPC or other on-premises networks via a VPN or Direct Connect. Typically the instances in the private subnet access the internet via a NAT gateway or NAT instance which is usually placed in the public subnet.**
- **Public subnets have a route to the internet via an Internet Gateway. Resources in public subnets can be accessed from the internet such as web servers whereas Private subnets do not have a direct route to the internet it uses NAT device to get connected with Internet. Resources within private subnets cannot be accessed from outside the VPC unless specific configurations are made.**

Route table: A route table contains a set of rules, called routes, that determine where network traffic from your subnet or gateway is directed. It is used to direct network traffic on a particular destination IP address. It allows customization of networks enabling them to connect to other subnets/gateways both within and outside of the VPC.

STEP1: LOGIN AS A ROOT USER AND SEARCH FOR VPC IN THE CONSOLE

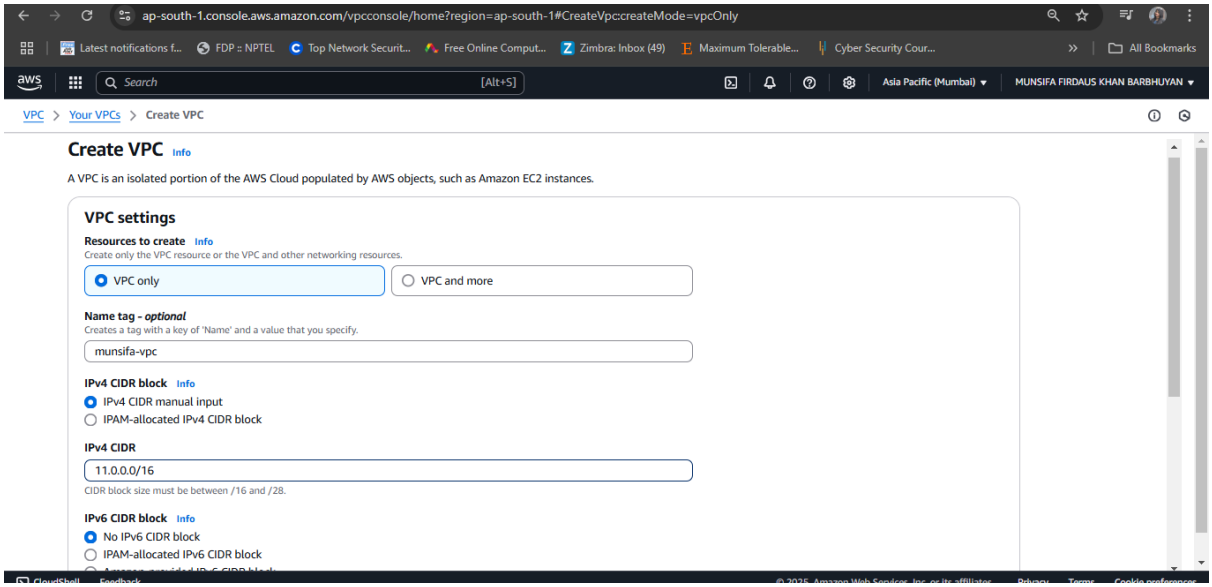


STEP2: CLICK ON CREATE VPC

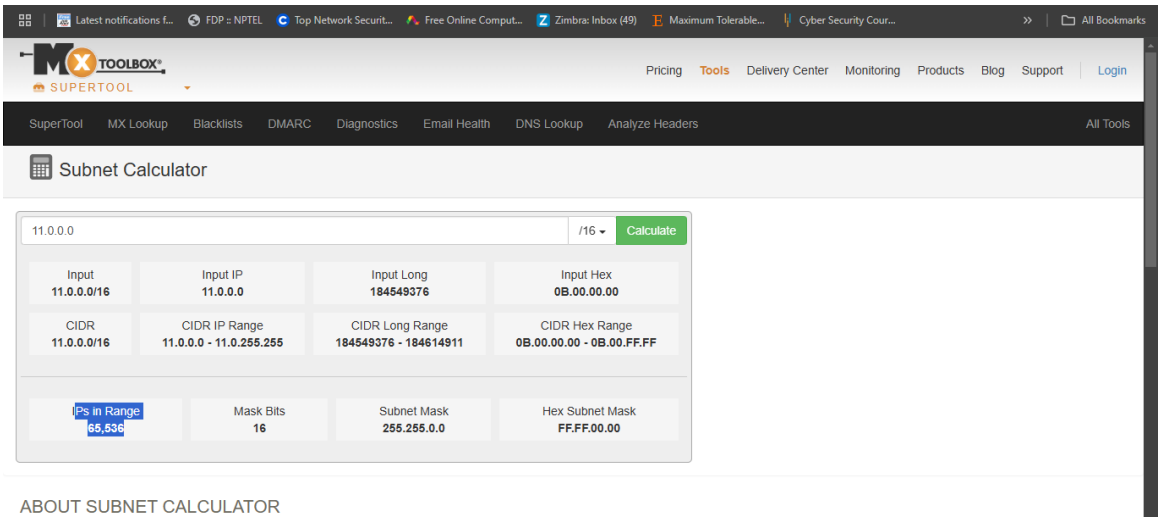


STEP3: SELECT VPC ONLY> TYPE THE VPC NAME> IPv4CIDR-11.0.0.0/16

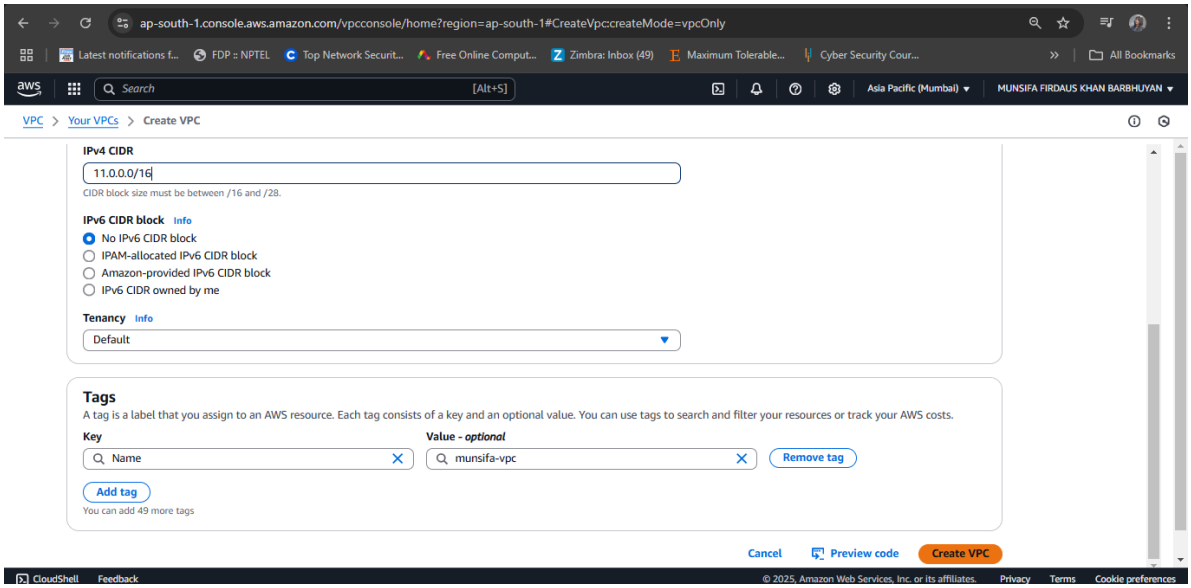
NOTE: In AWS a VPC spans a specific IP address range using CIDR (Classless Inter-Domain Routing) blocks. The CIDR block defines the range of IP addresses that can be assigned to resources within the VPC



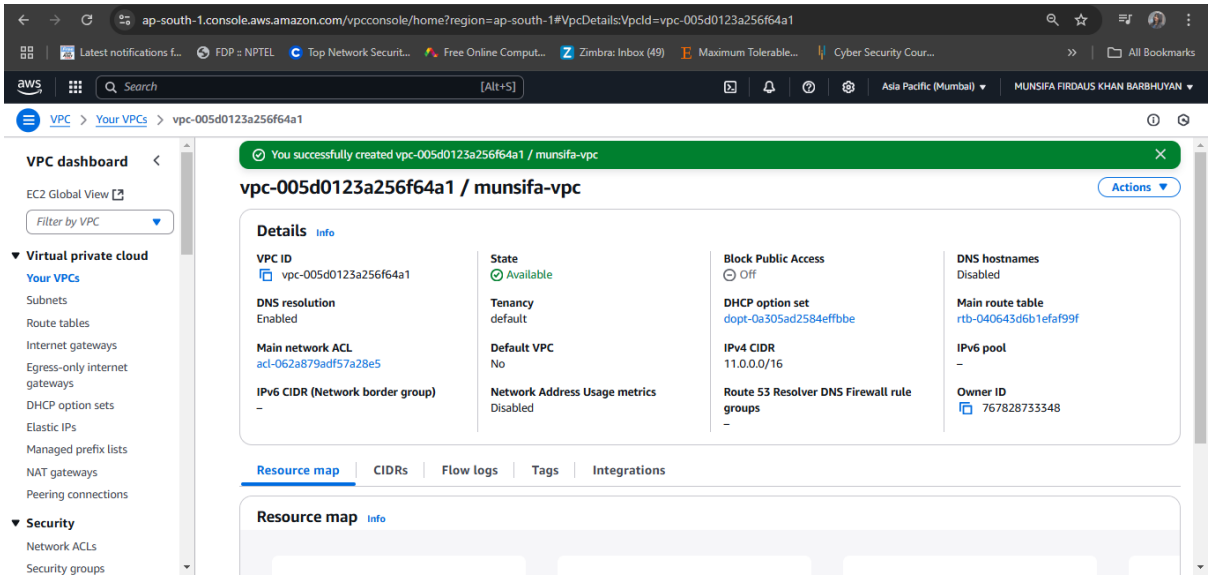
For better understanding of the CIDR range, Search CIDR Range Calculator in google and enter the range. Here its showing 65536 IP range, i.e. these are the maximum number of IP addresses which is accessible by the VPC. If we increase from /16 to /32, it will decrease the number of IP address and if we decrease it to /8, it will increase the number of IP addresses. (/16 means first 16bits are locked)



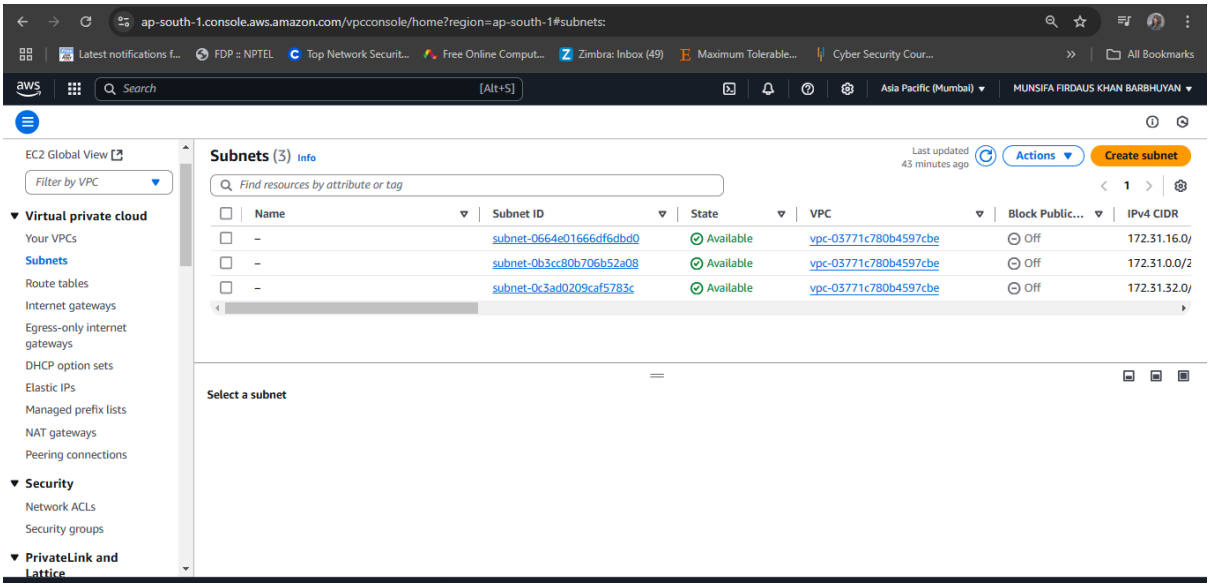
STEP4: CLICK ON CREATE VPC



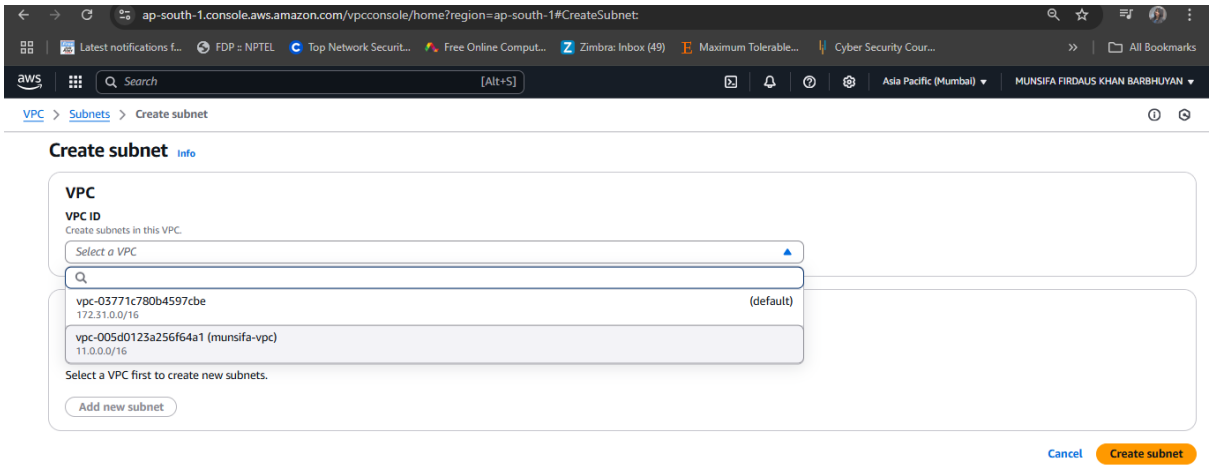
STEP5: VPC CREATED



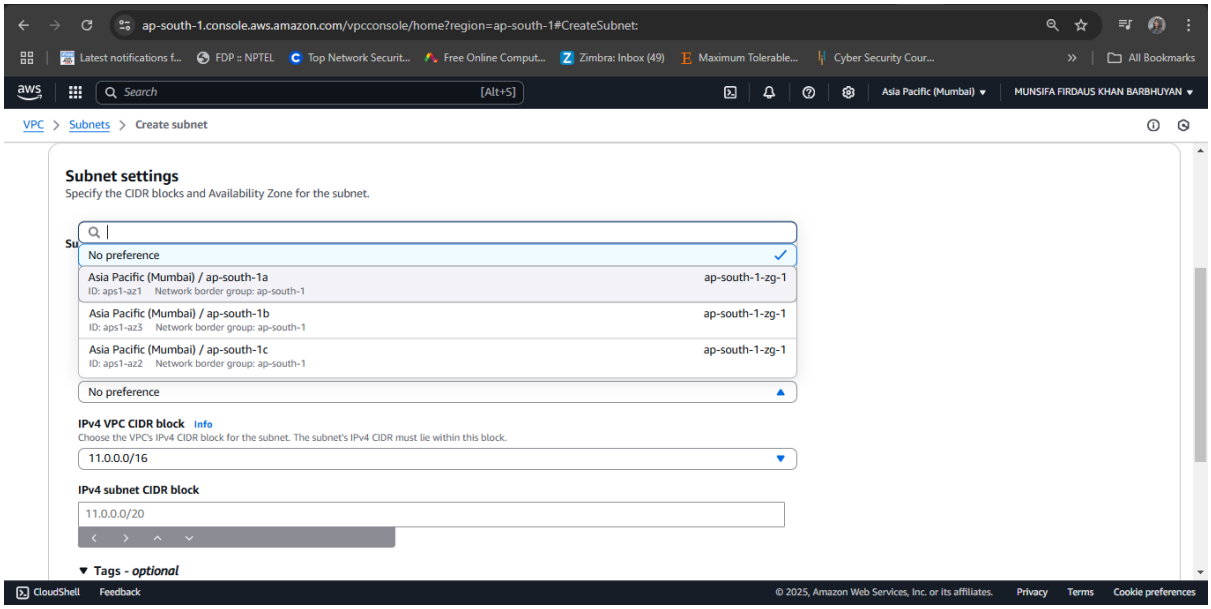
STEP6: CLICK ON SUBNET>CREATE SUBNET



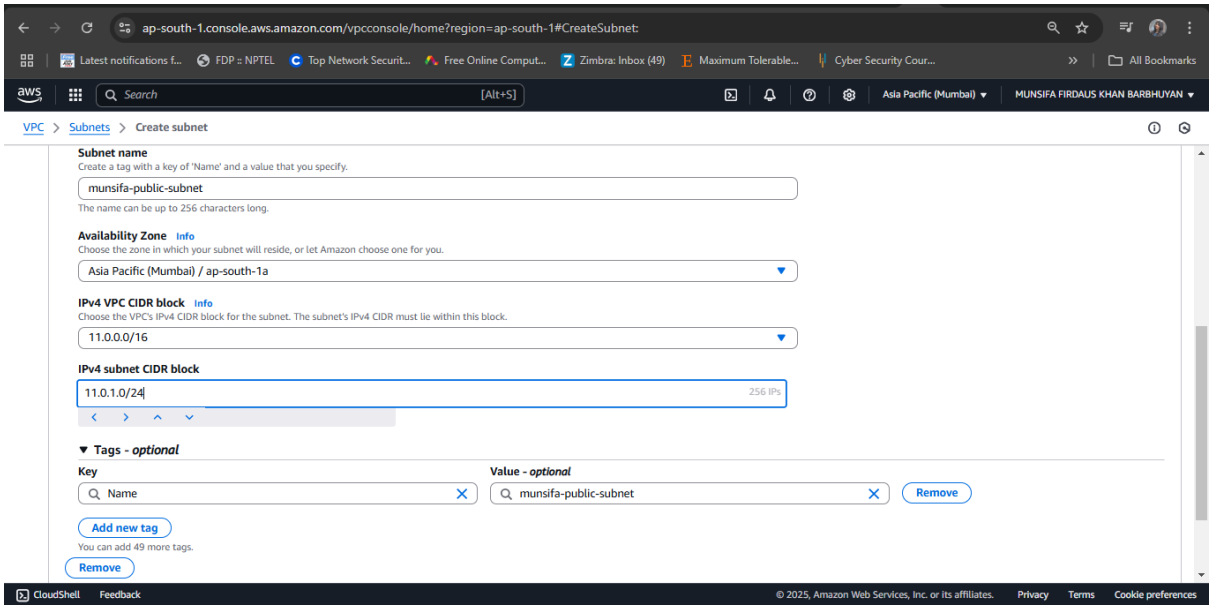
STEP7: SELECT THE VPC THAT YOU HAVE CREATED EARLIER



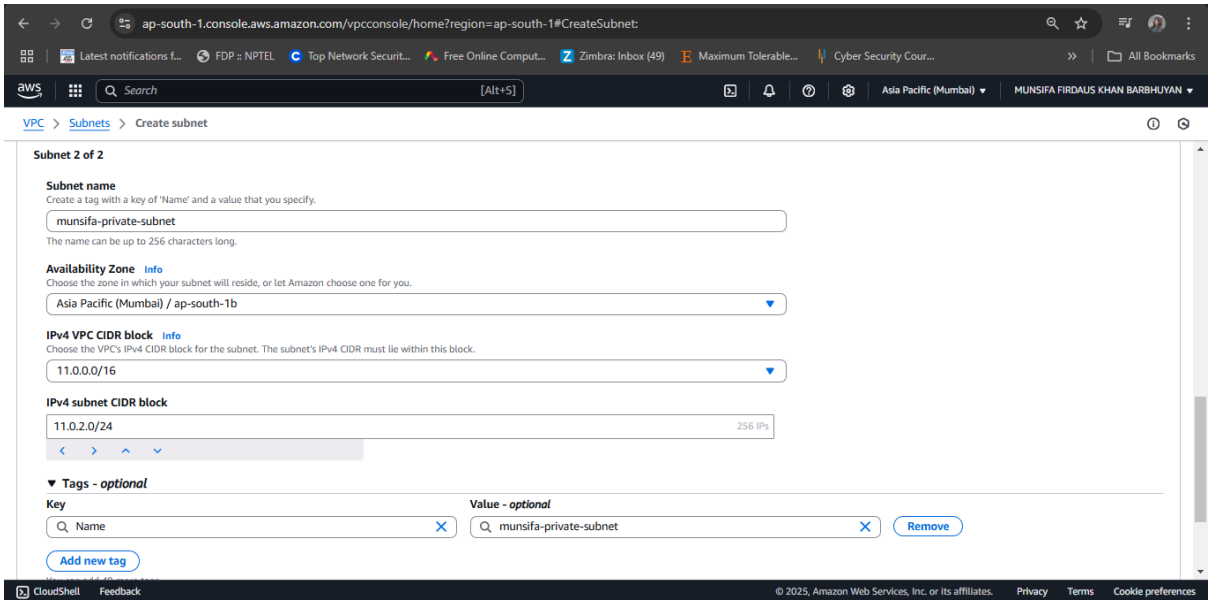
STEP8: SELECT ap-south-1a in the Availability Zone



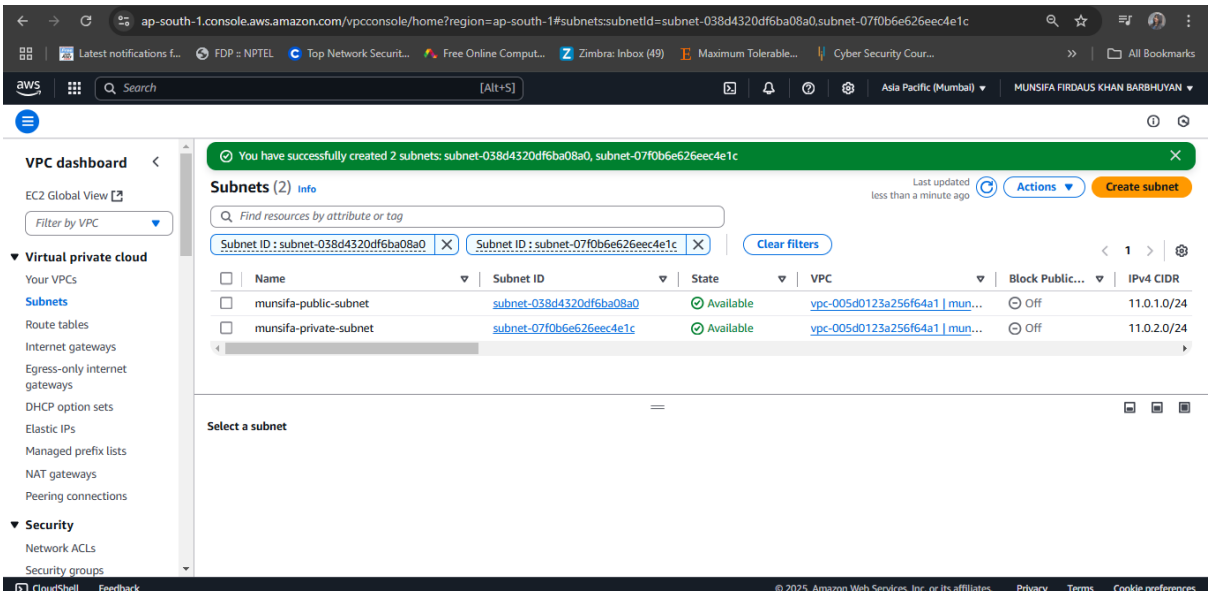
STEP9: ENTER IPv4 SUBNET CIDR BLOCK-11.0.1.0/24> CLICK ON ADD NEW SUBNET



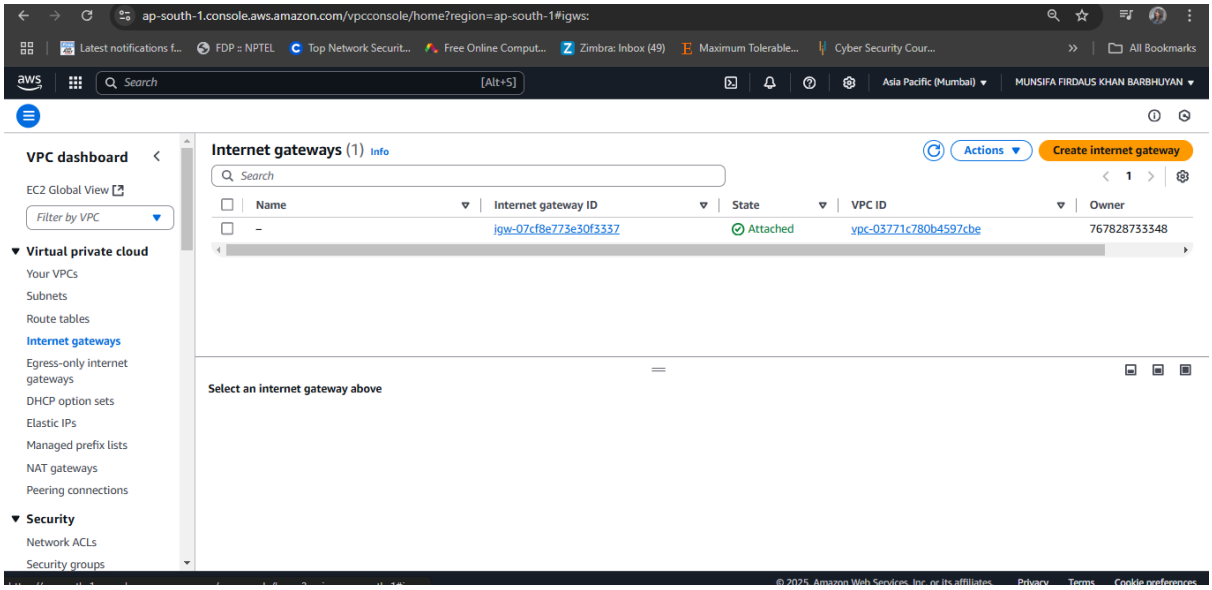
STEP9: ENTER THE NAME OF THE PRIVATE SUBNET> AVAILABILITY ZONE:ap-south-1b> IPv4 subnet CIDR Block-11.0.2.0/24> CLICK ON CREATE SUBNET



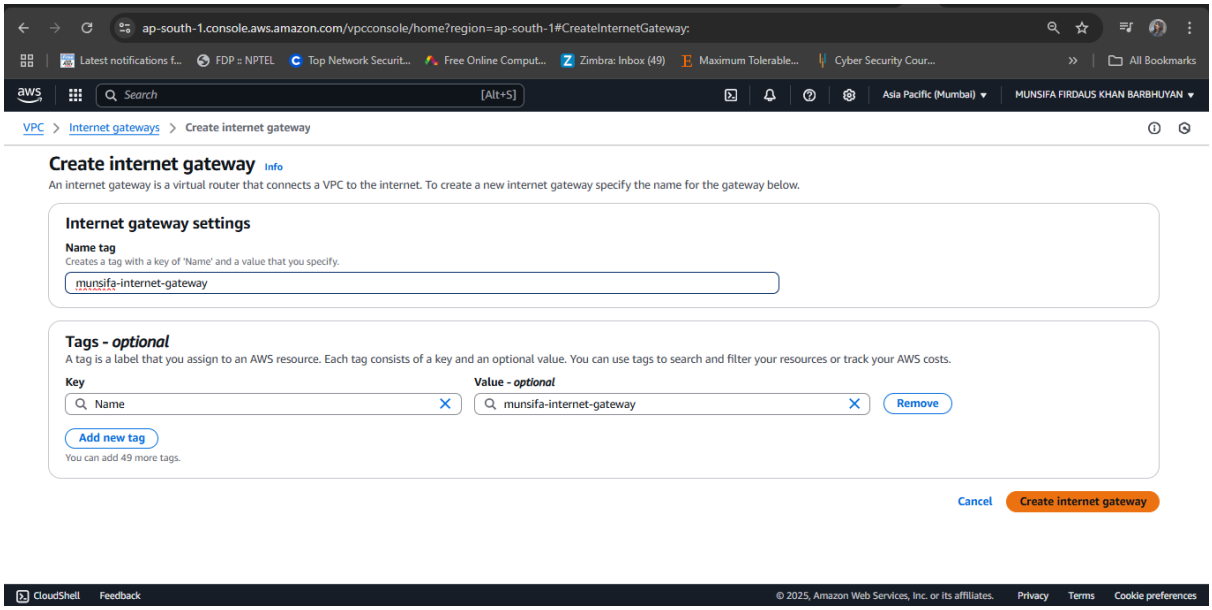
STEP10: BOTH THE PUBLIC AND PRIVATE SUBNET IS CREATED



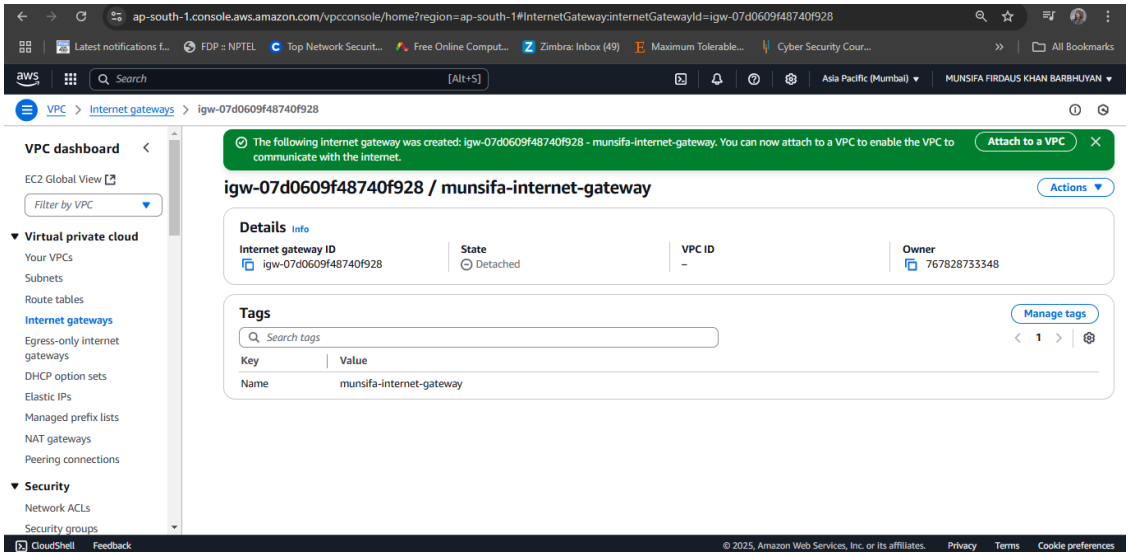
STEP11: NOW CREATE AN INTERNET GATEWAY FOR PUBLIC SUBNET. CLICK ON INTERNET GATEWAY FROM THE DASHBOARD> CLICK ON CREATE INTERNET GATEWAY



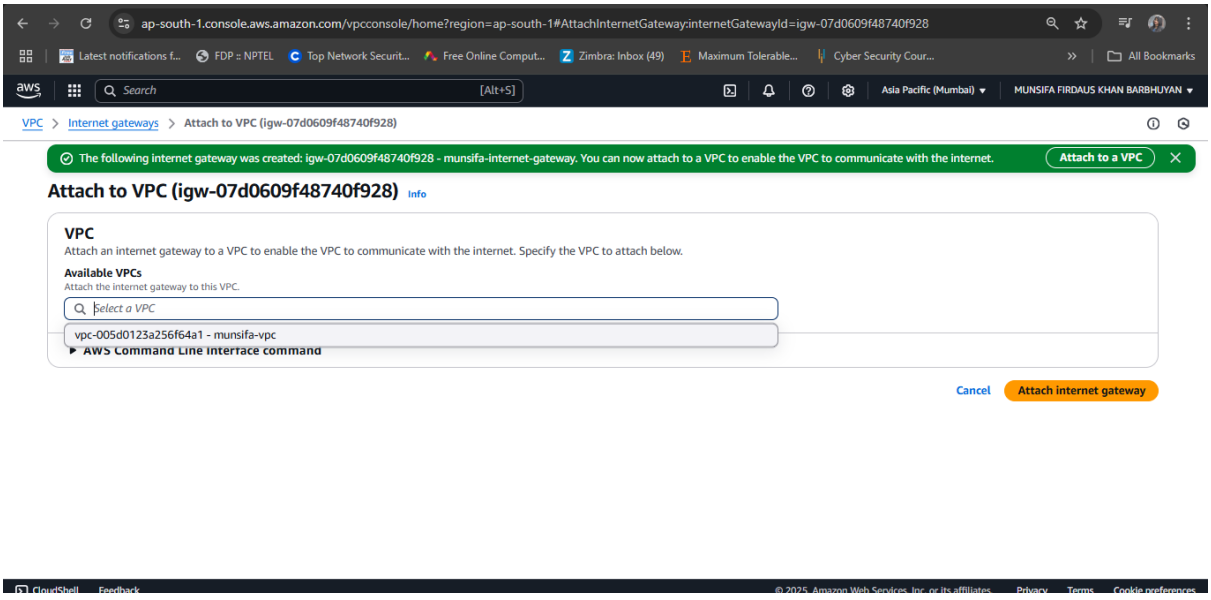
STEP12: TYPE THE NAME OF THE GATEWAY



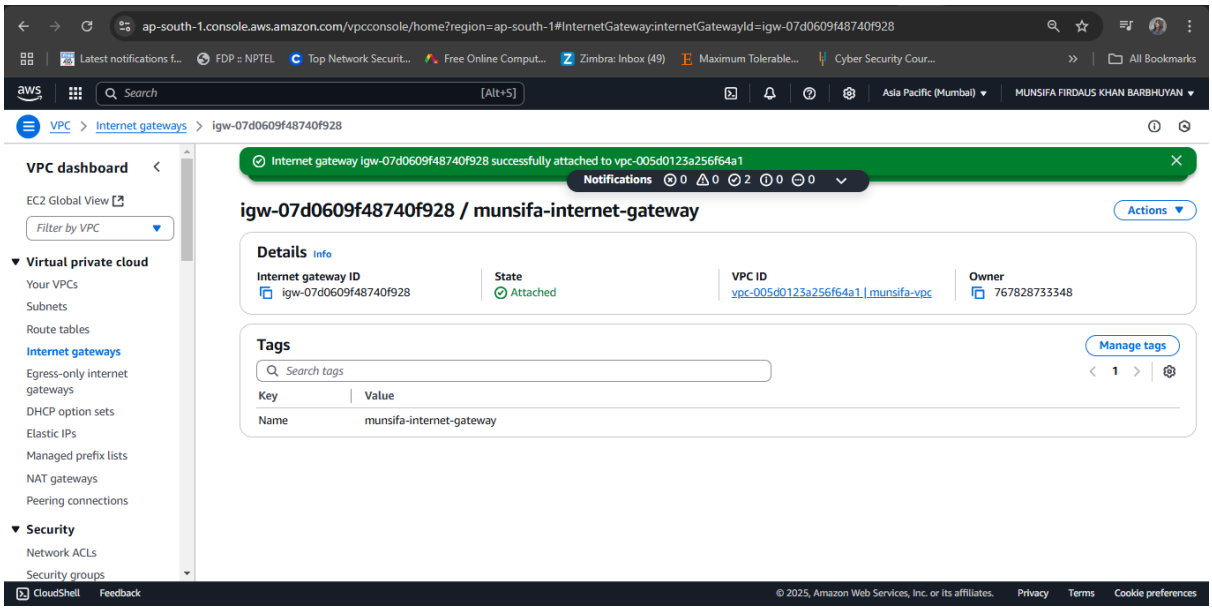
STEP13: INTERNET GATEWAY IS CREATED



STEP14: CLICK ON ATTACH TO VPC> SELECT THE VPC CREATED EARLIER> CLICK ON ATTACH INTERNET GATEWAY

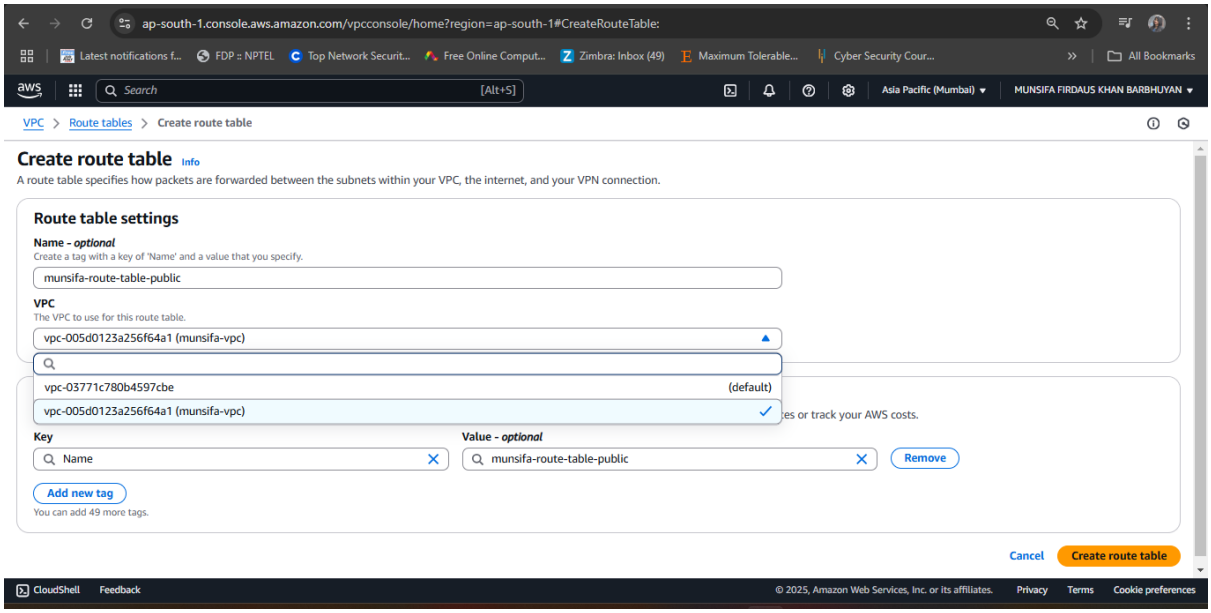


STEP15: INTERNET GATEWAY IS ATTACHED TO THE VPC

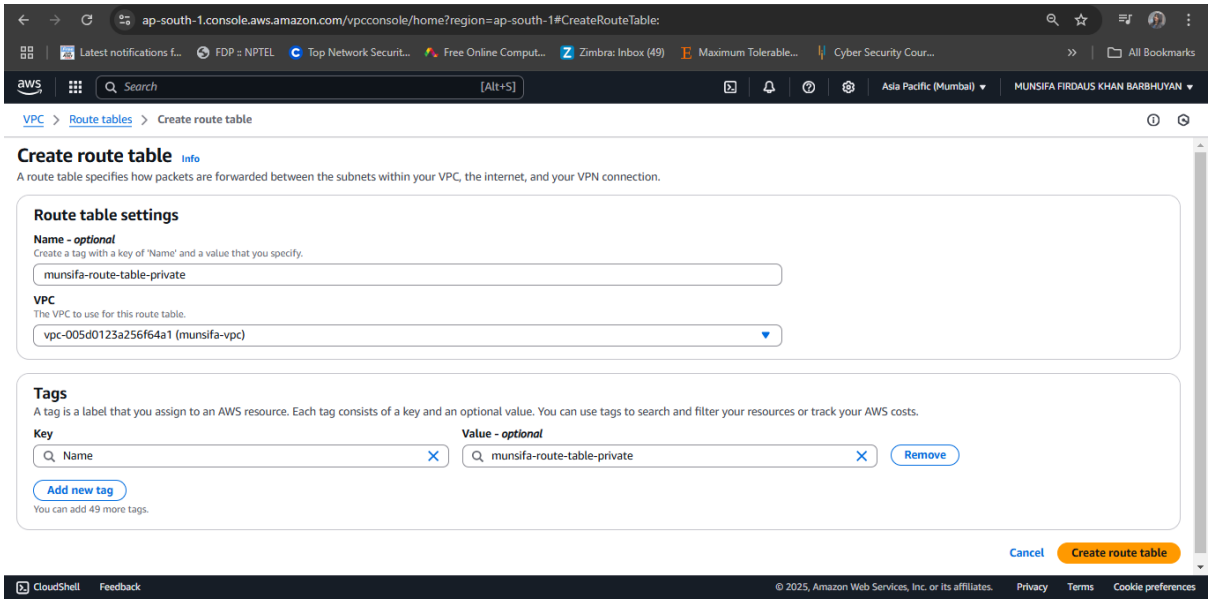


STEP16: NEXT CREATE TWO ROUTE TABLES (PRIVATE AND PUBLIC) FOR THE SUBNETS.

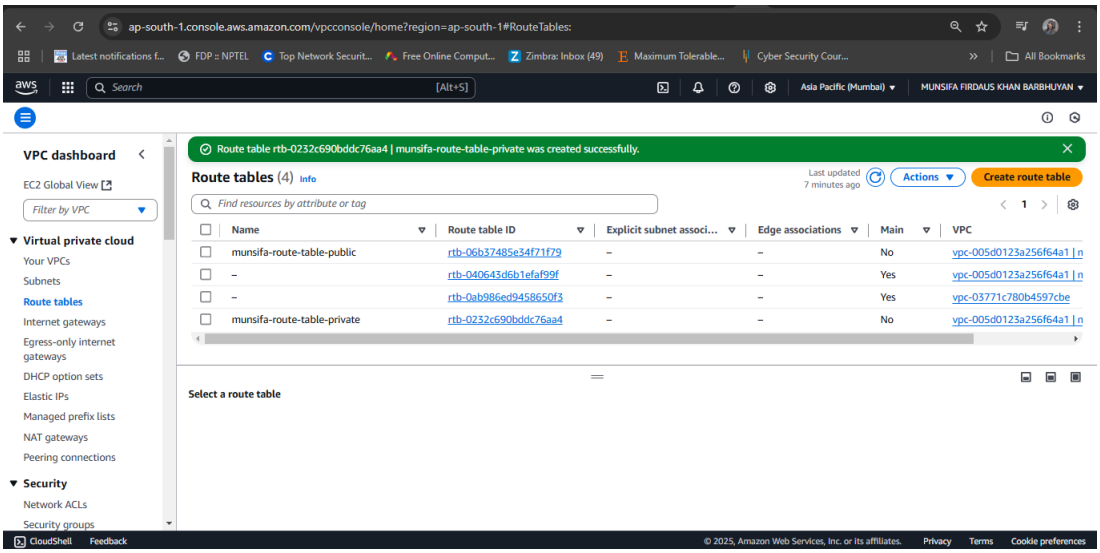
STEP17: TO CREATE A PUBLIC ROUTE TABLE: CLICK ON ROUTE TABLES>CREATE A ROUTE TABLE> WRITE A SUITABLE NAME FOR THE ROUTE TABLE>CHOOSE THE VPC>CLICK ON CREATE ROUTE TABLE



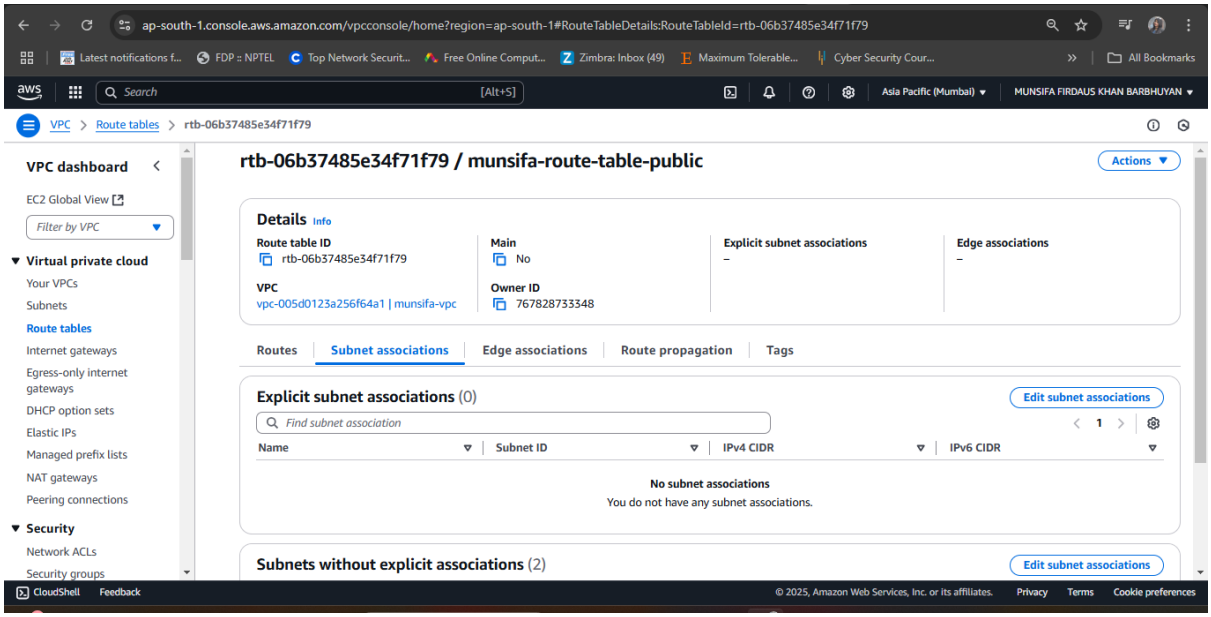
STEP18: TO CREATE A PRIVATE ROUTE TABLE: CLICK ON ROUTE TABLES>CREATE A ROUTE TABLE> WRITE A SUITABLE NAME FOR THE ROUTE TABLE>CHOOSE THE VPC>CLICK ON CREATE ROUTE TABLE



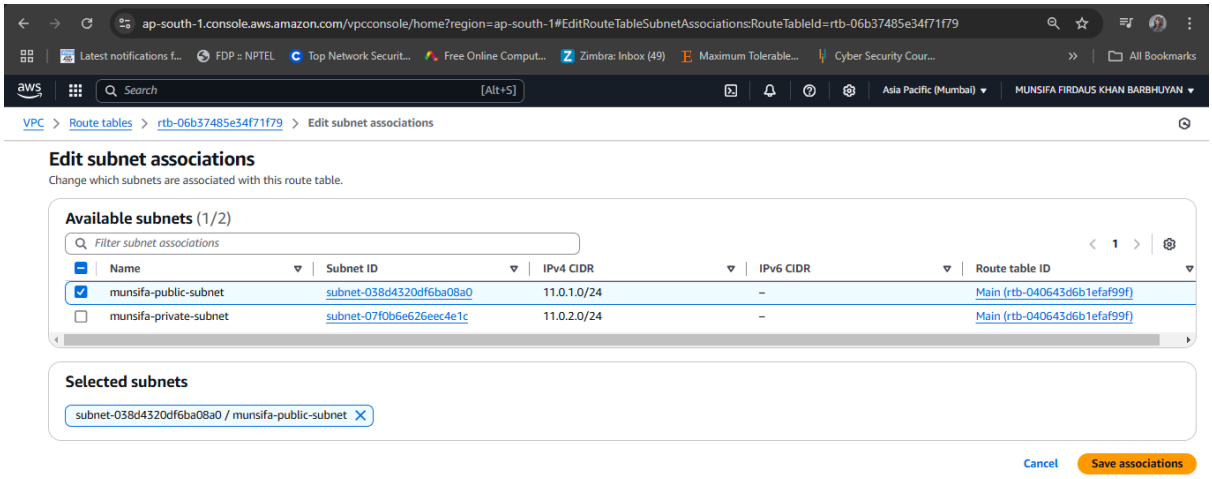
STEP19: BOTH THE PUBLIC AND PRIVATE ROUTE TABLES ARE CREATED AS SHOWN BELOW. NOW WE NEED TO CONNECT BOTH THE ROUTE TABLES WITH THEIR CORRESPONDING SUBNETS.



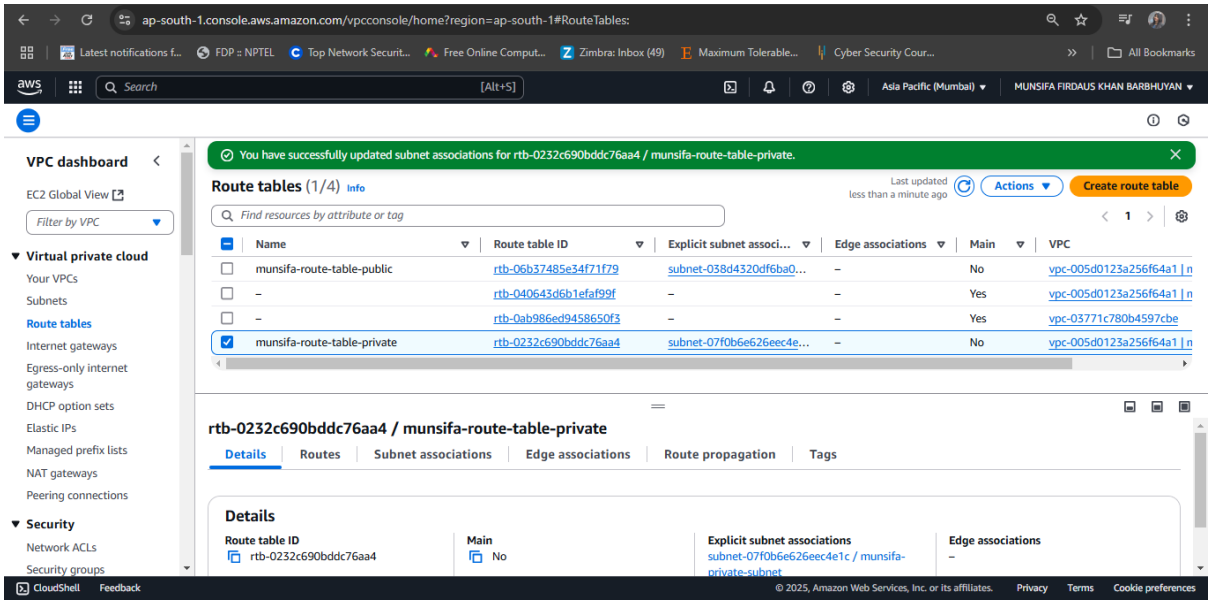
STEP20: CLICK ON THE PUBLIC ROUTE TABLE>CLICK ON SUBNET ASSOCIATIONS



STEP 21: CLICK ON EDIT SUBNET ASSOCIATIONS > CHOOSE PUBLIC SUBNET FROM THE OPTIONS> CLICK ON SAVE ASSOCIATIONS

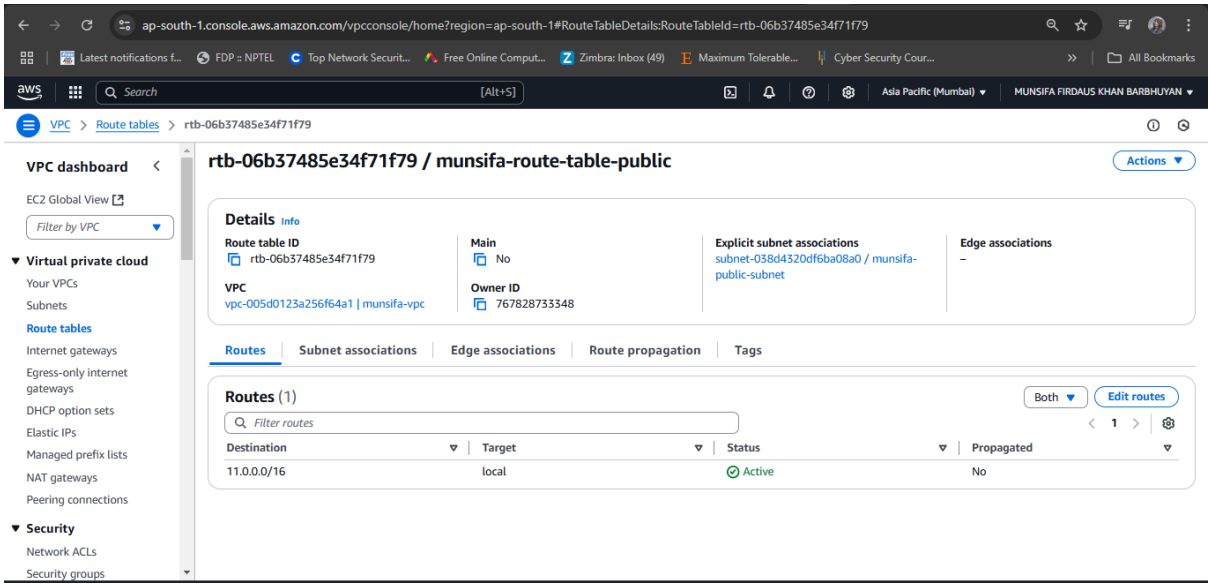


STEP 21: SUBNET ASSOCIATION IS DONE. REPEAT THE SAME STEP FOR PRIVATE SUBNET ASSOCIATION.

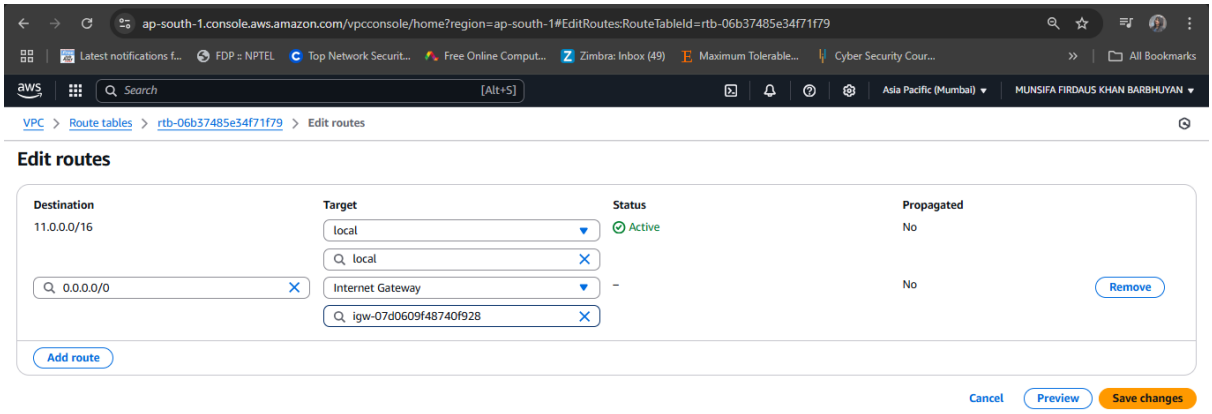


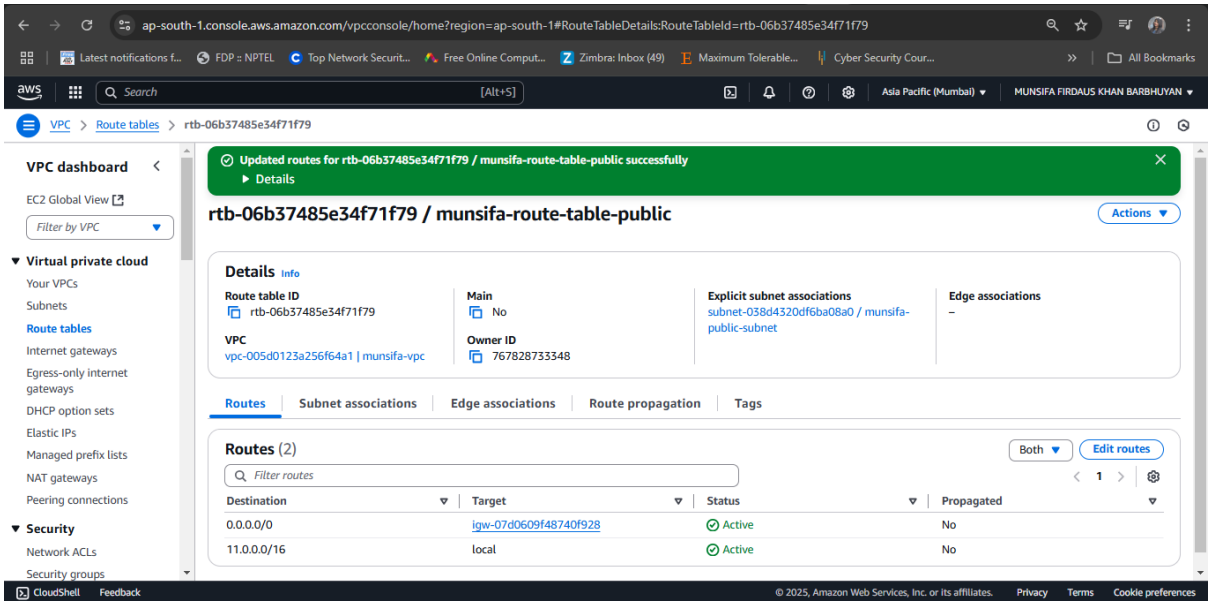
STEP22: NEXT WE NEED TO CREATE THE ROUTE SO THAT INTERNET CAN BE ACCESSED WITH THE HELP OF INTERNET GATEWAY THROUGH THESE ROUTE TABLES.

STEP23: GO TO PUBLIC ROUTE TABLE FROM THE AWS CONSOLE> CLICK ON ROUTES>CLICK ON EDIT ROUTES



STEP24: CLICK ON ADD ROUTE>SELECT THE IP-0.0.0.0/0>CHOOSE INTERNET GATEWAY> SELECT THE INTERNET GATEWAY THAT YOU HAVE CREATED> CLICK ON SAVE CHANGES (NOTE:0.0.0.0/0 means it allows all the IP addresses to access the resources present in the public subnet)



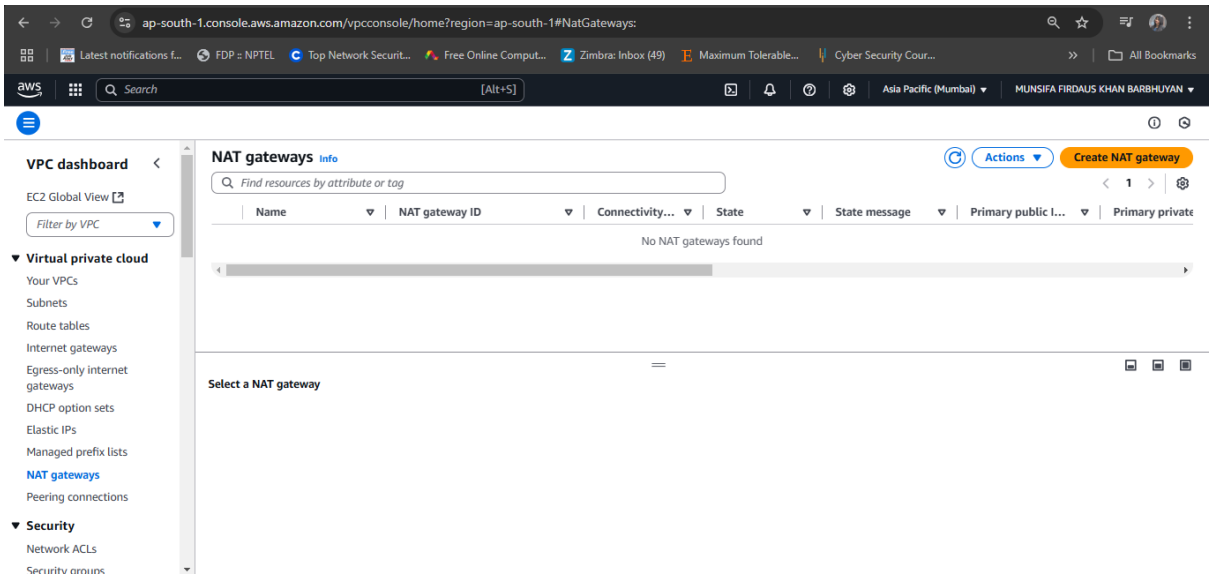


STEP25: WE NEED TO CREATE THE NAT GATEWAY FOR THE PRIVATE SUBNET SO THAT RESOURCES PRESENT INSIDE THE PRIVATE SUBNET CAN ACCESS THE INTERNET WITH THE HELP OF INTERNET GATEWAY.

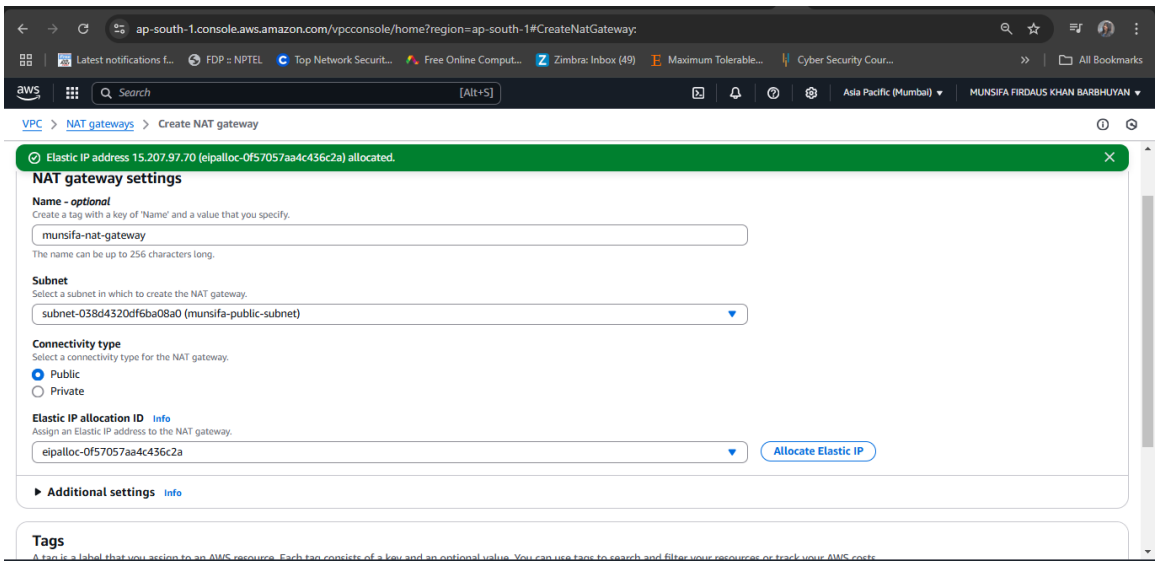
NOTE:

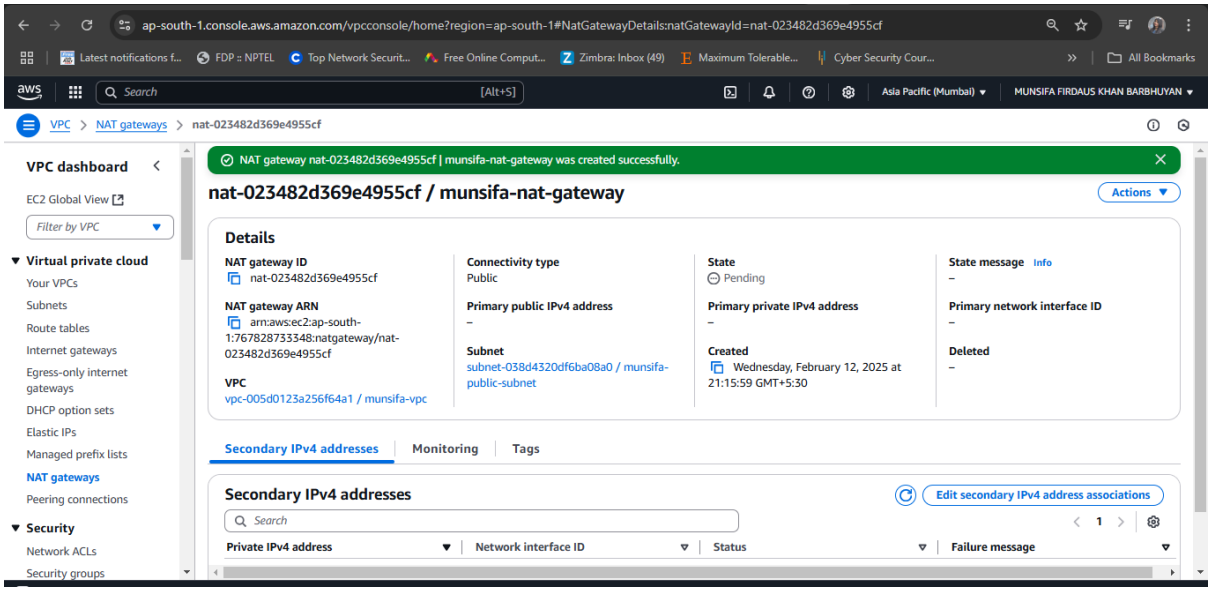
1. **Internet Gateway is two-way process i.e., outside world can access the resources inside the public subnet as well as resources inside it can request for accessing outside.**
2. **NAT GATEWAY is not free and its created only in public subnet.**
3. **Private subnet will be able to access the NAT gateway with the help of its route table to access resources inside the public subnet.**

STEP26: CLICK ON NAT GATEWAYS FROM THE DASHBOARD> CLICK ON CREATE NAT GATEWAY.

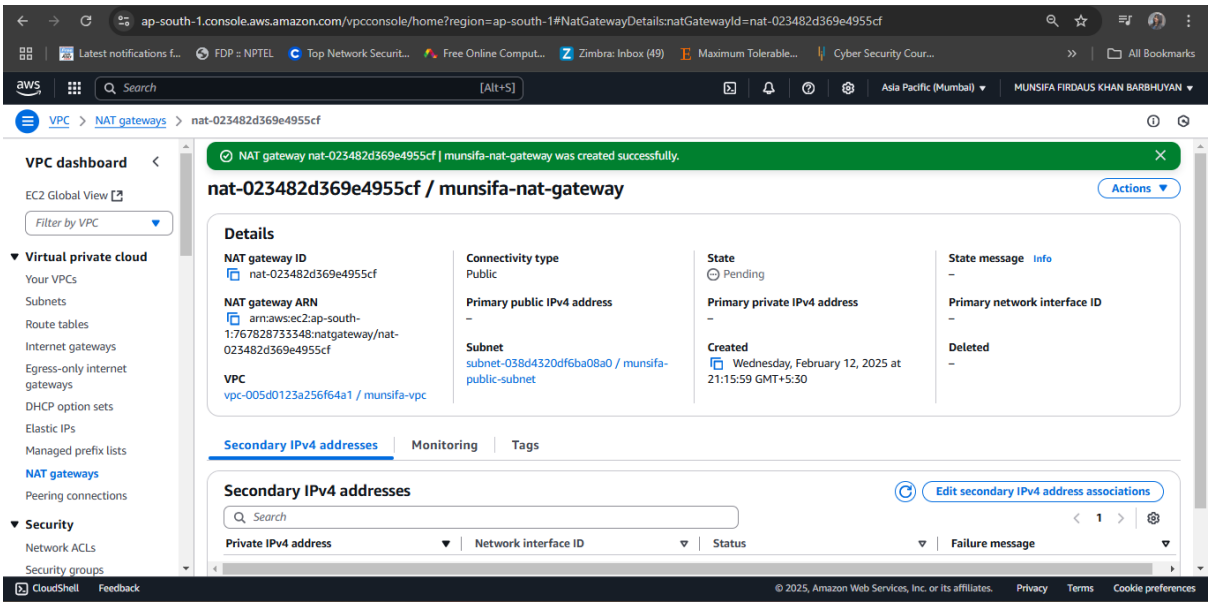


STEP27: GIVE A NAME TO THE NAT GATEWAY>CHOOSE THE PUBLIC SUBNET>ALLOCATE ELASTIC IP> CLICK ON CREATE NAT GATEWAY.



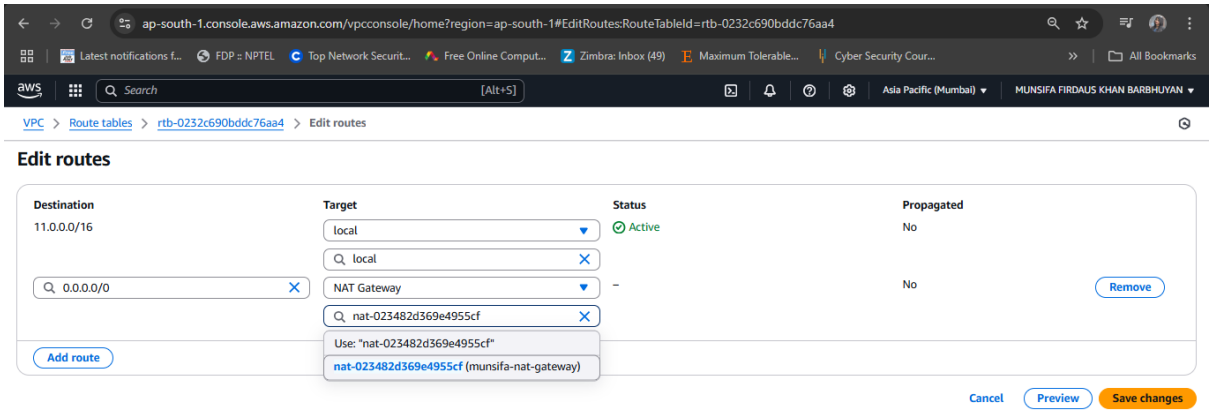


STEP28: NOW UPDATE THE PRIVATE ROUTE TABLE: CLICK ON THE ROUTE TABLES>CLICK ON THE PRIVATE ROUTE TABLE>CLICK ON ROUTES>CLICK ON EDIT ROUTES

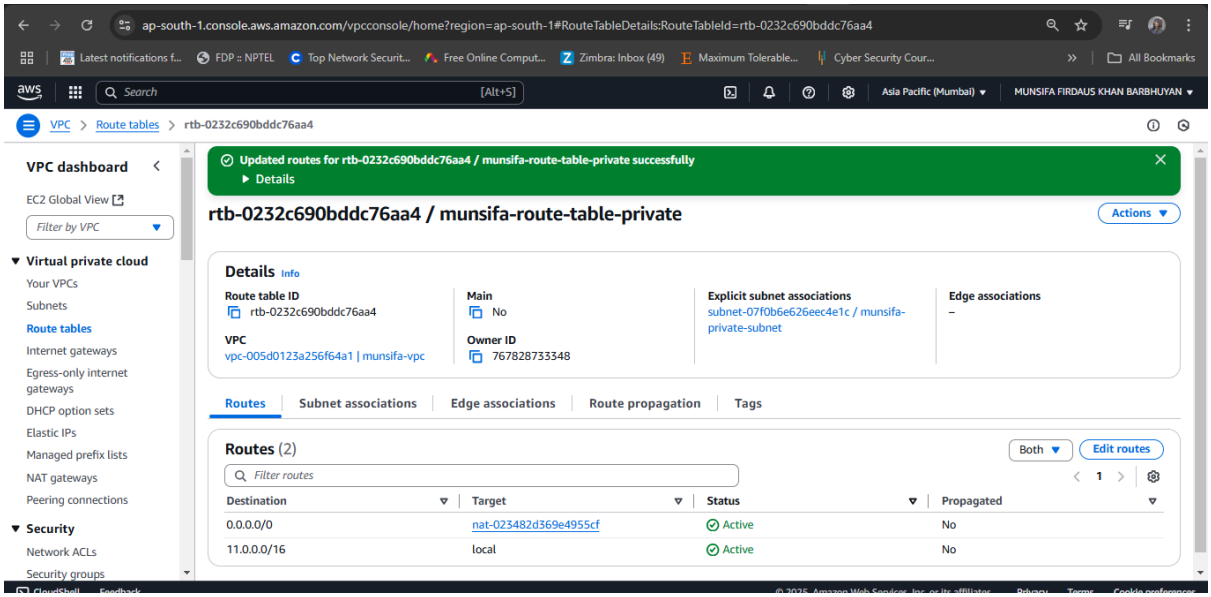


STEP29: CLICK ON ADD ROUTE>SELECT IP ADDRESS

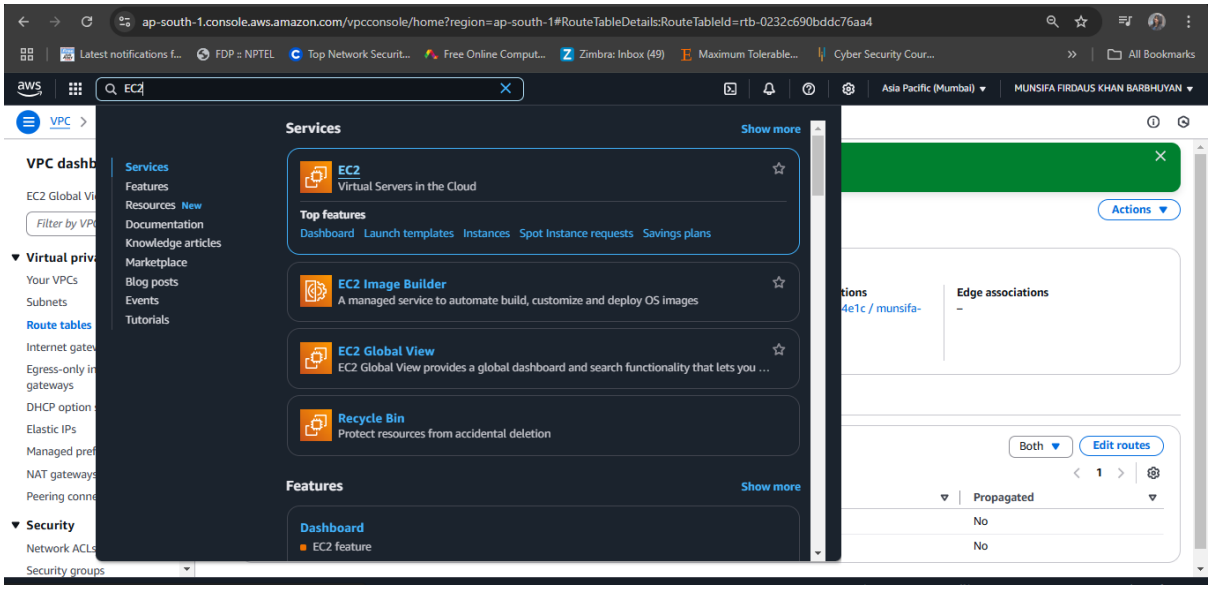
:0.0.0.0/0>SELECT NAT GATEWAY>CHOOSE THE NAT GATEWAY THAT YOU HAVE CREATED> CLICK ON SAVE CHANGES.



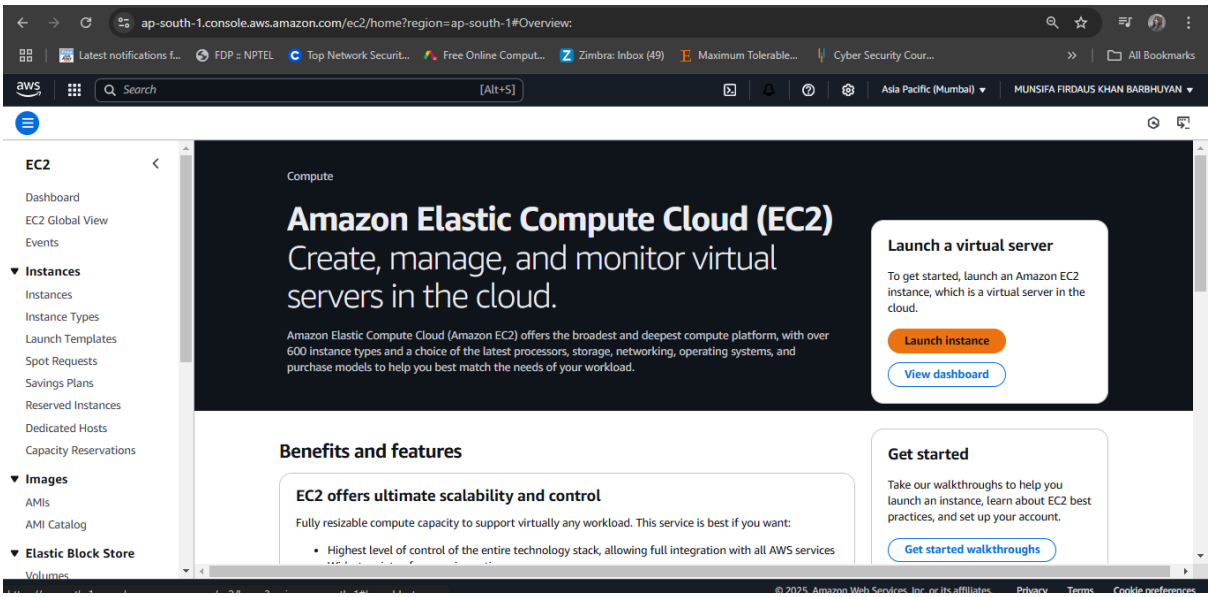
STEP30: PRIVATE SUBNET IS UPDATED WITH NAT GATEWAY



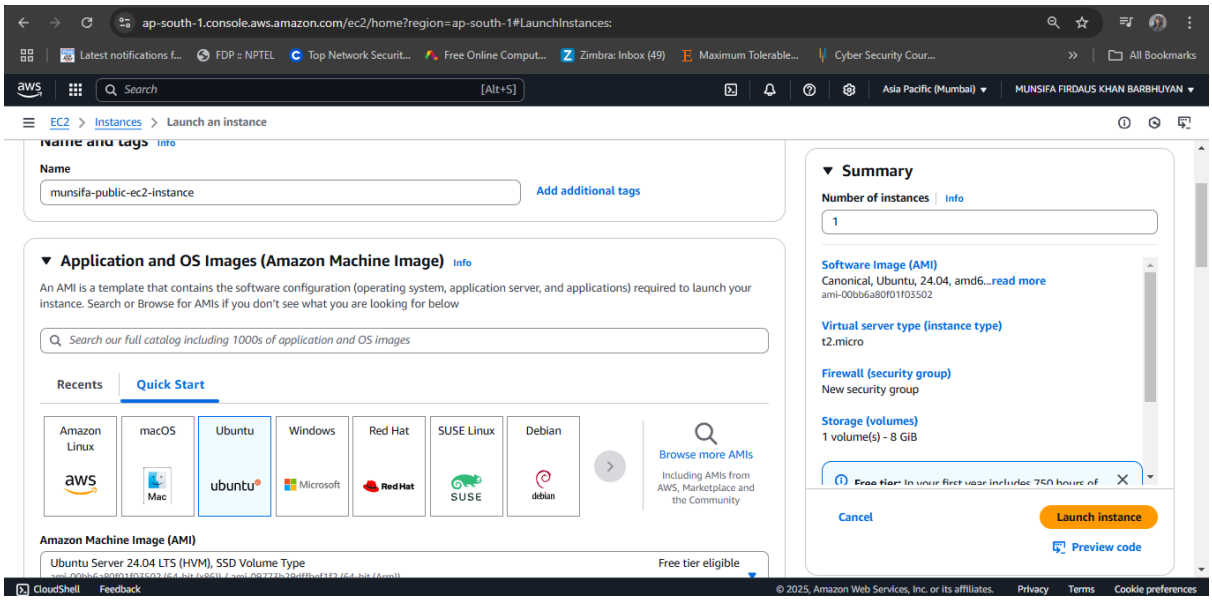
STEP31: WHOLE NETWORK SET UP IS READY, NOW WE HAVE TO CREATE EC2 INSTANCES FOR BOTH PUBLIC AND PRIVATE SUBNET. SEARCH FOR EC2>CLICK ON EC2



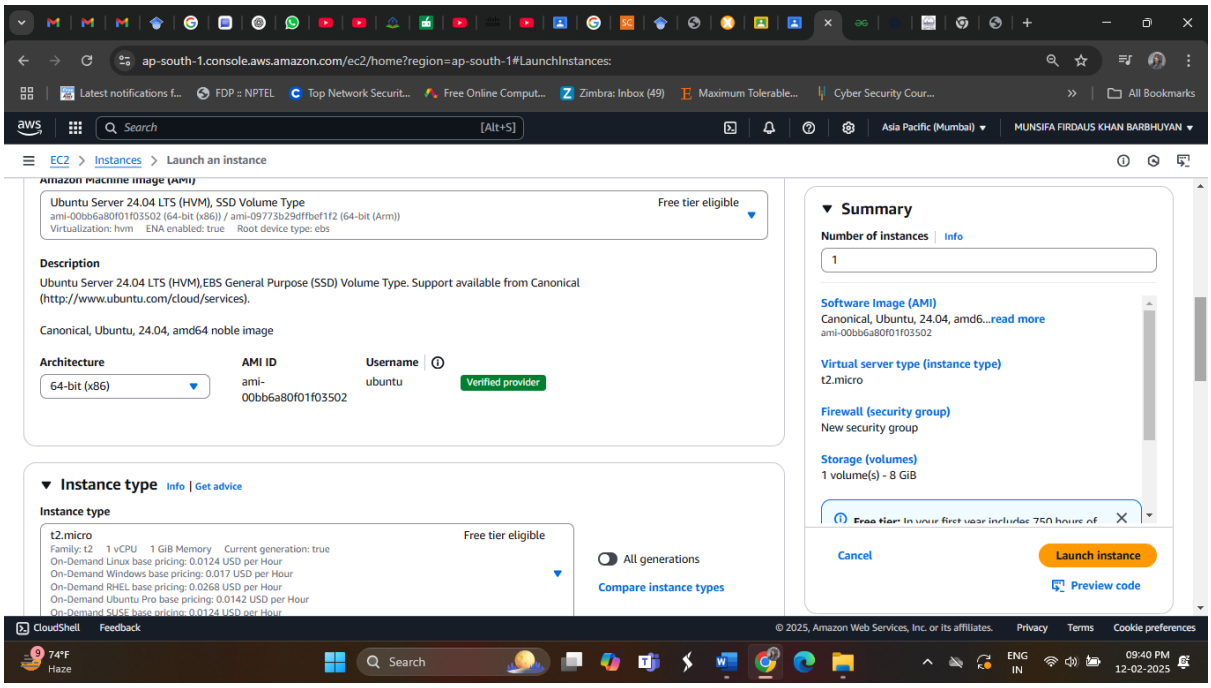
STEP 32: CLICK ON LAUNCH INSTANCE



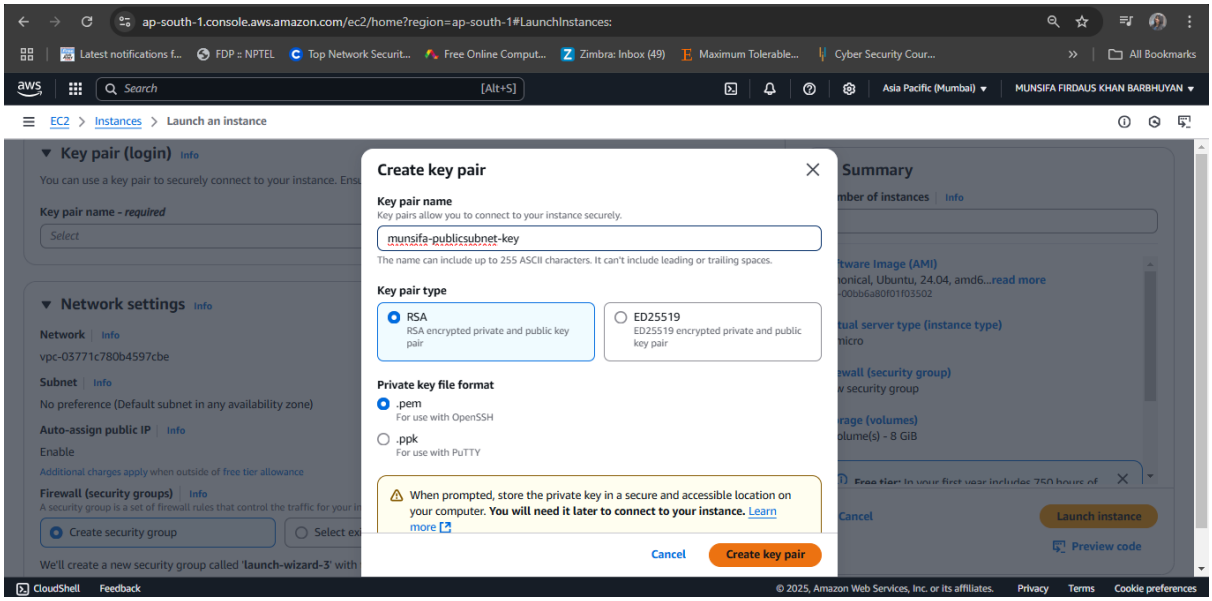
STEP 33: GIVE A SUITABLE INSTANCE NAME>CHOOSE OS:UBUNTU



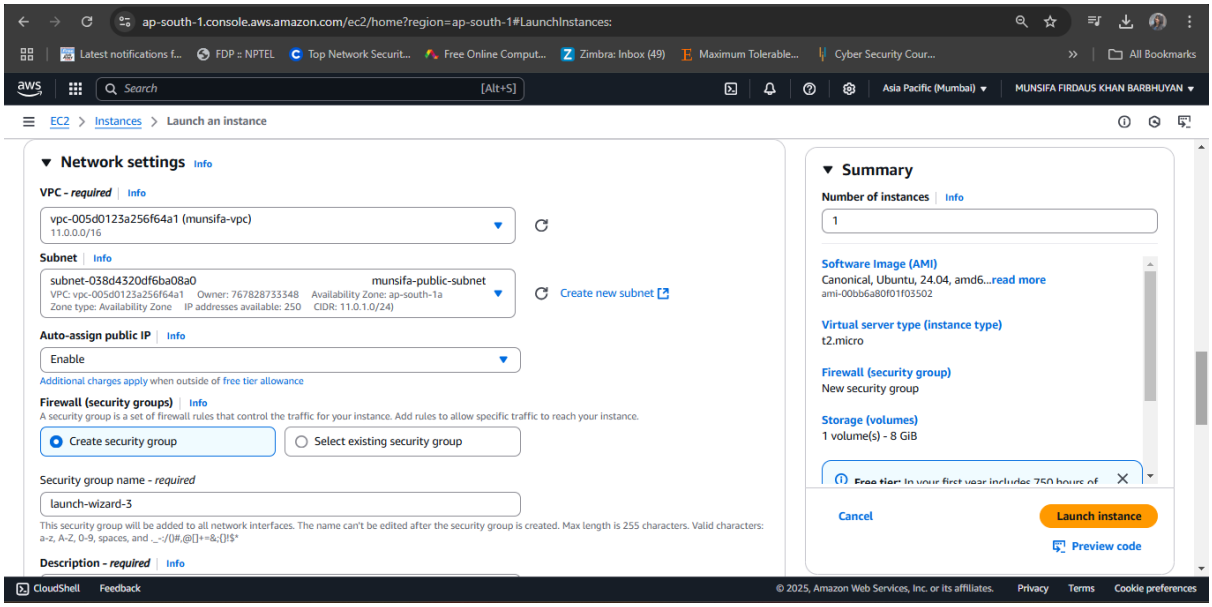
STEP 34: CHOOSE THE DEFAULT ARCHITECTURE> FREE TIER INSTANCE TYPE



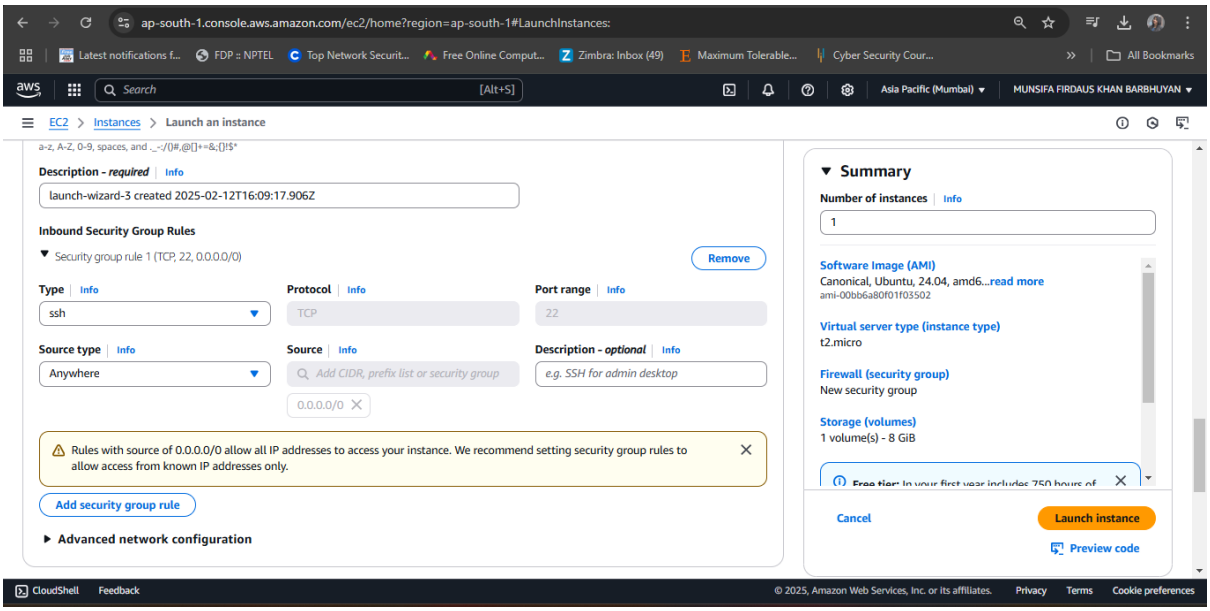
STEP 35: CREATE KEY PAIR



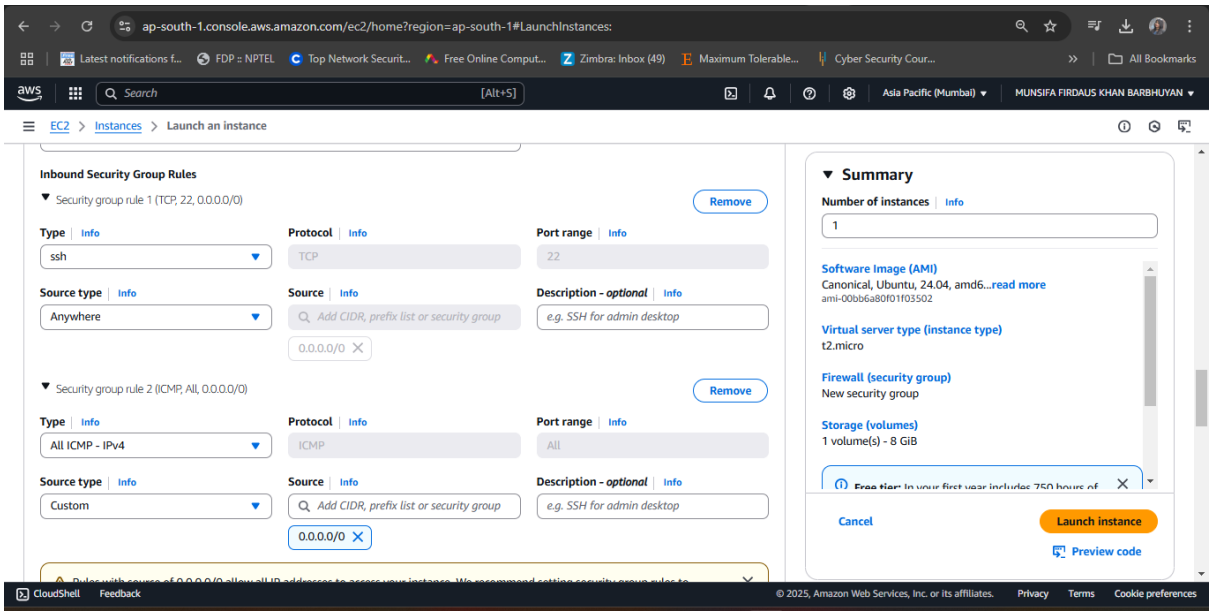
STEP 36: EDIT THE NETWORK SETTING>CHOOSE THE VPC CREATED EARLIER>SELECT THE PUBLIC SUBNET> ENABLE THE AUTO-ASSIGN PUBLIC IP>CHOOSE CREATE SECURITY GROUP



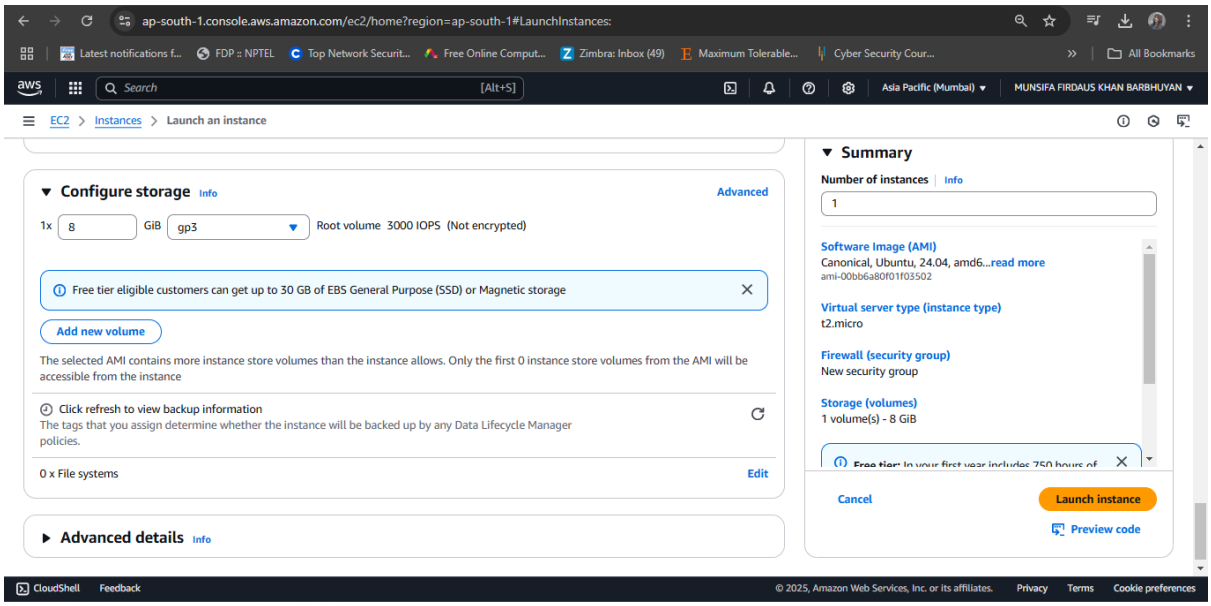
STEP 37: SECURITY GROUP RULE-1 IS DEFAULT> CLICK ON ADD SECURITY GROUP RULE



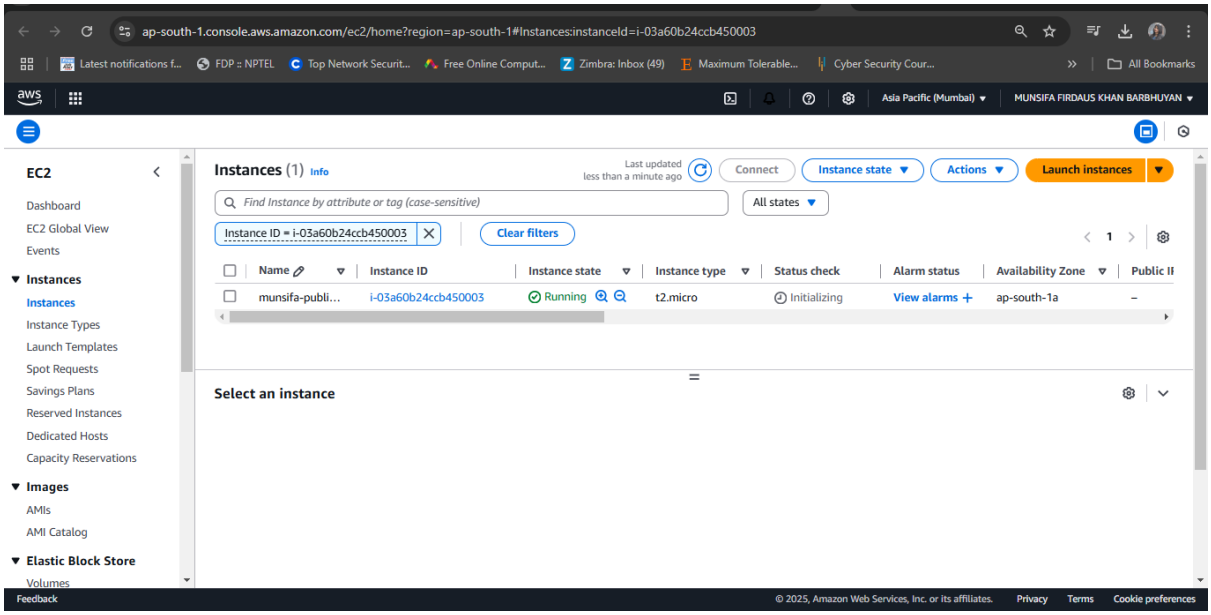
STEP 38: CHOOSE ALL ICMP-IPV4> SOURCE:0.0.0.0/0



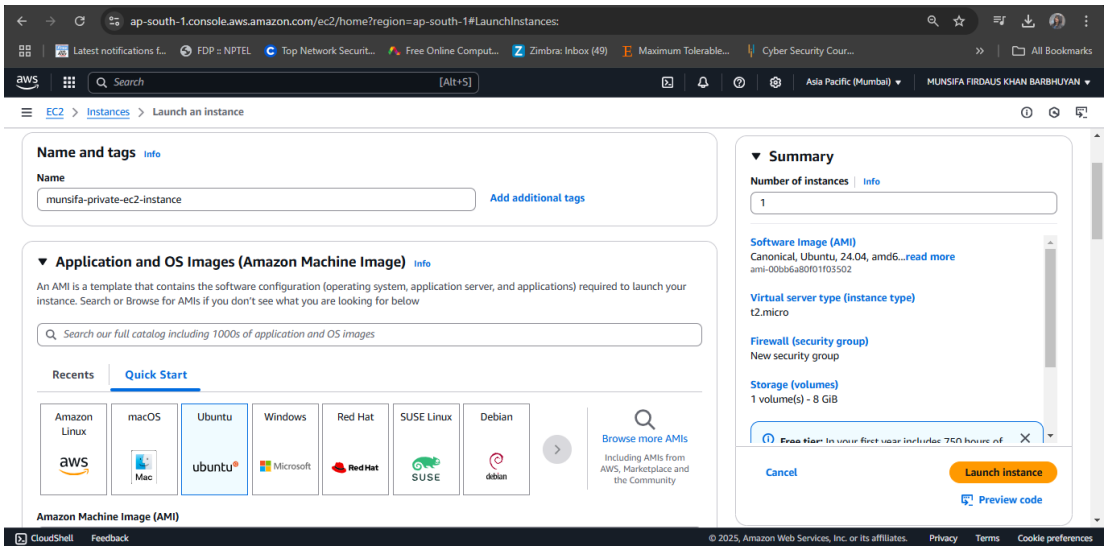
STEP 39: CHOOSE DEFAULT CONFIGURE STORAGE: 8GB>CLICK ON LAUNCH INSTANCE

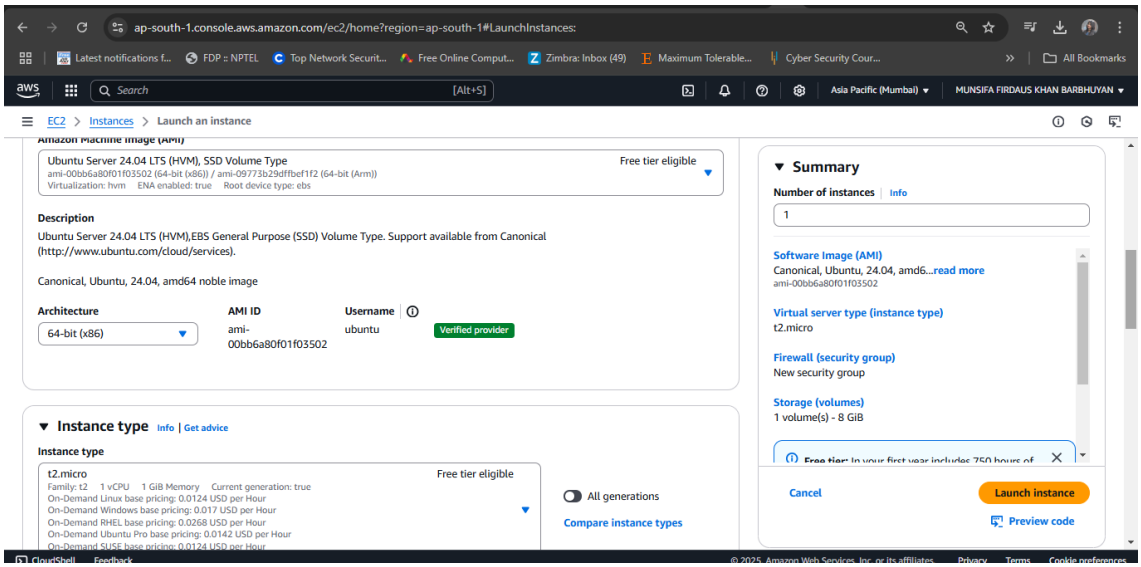


STEP 40: PUBLIC INSTANCE CREATED

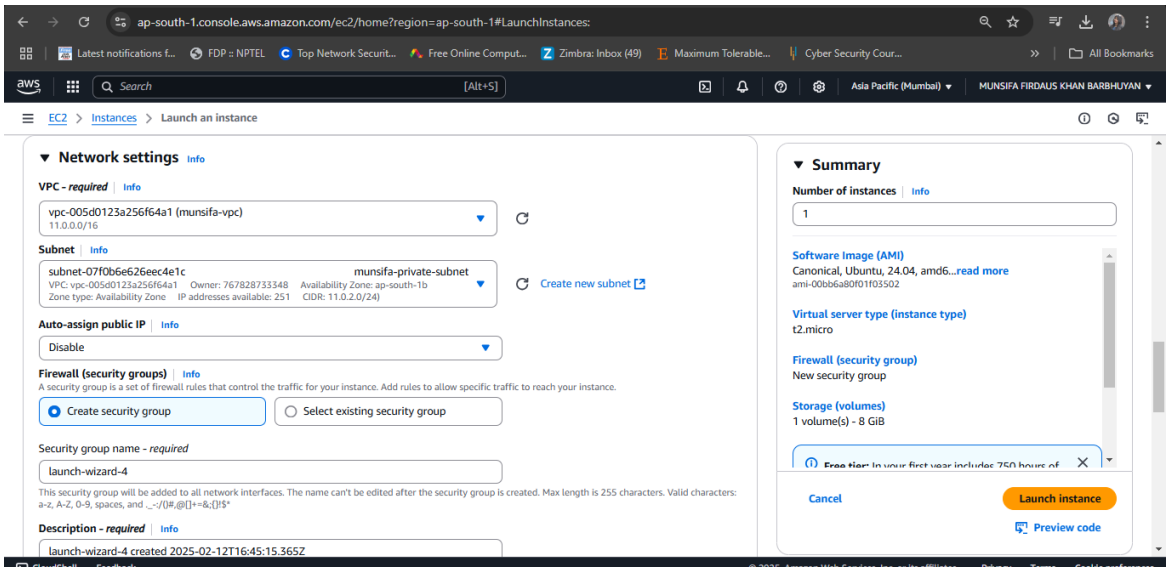


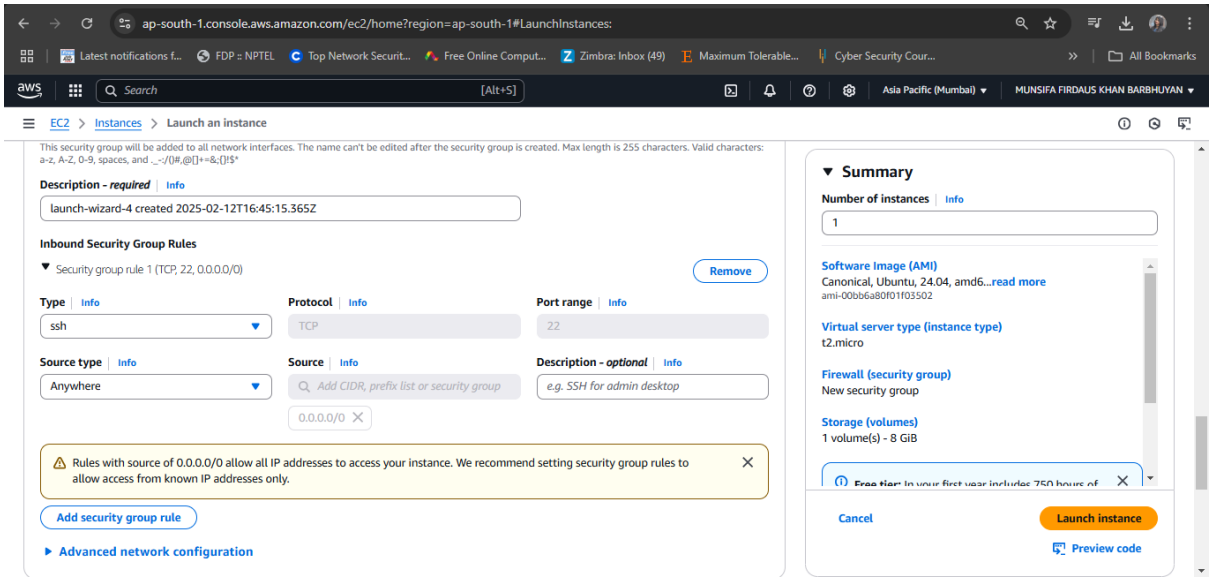
STEP 41: CREATE EC2 INSTANCE FOR PRIVATE SUBNET WITH THE SAME STEPS



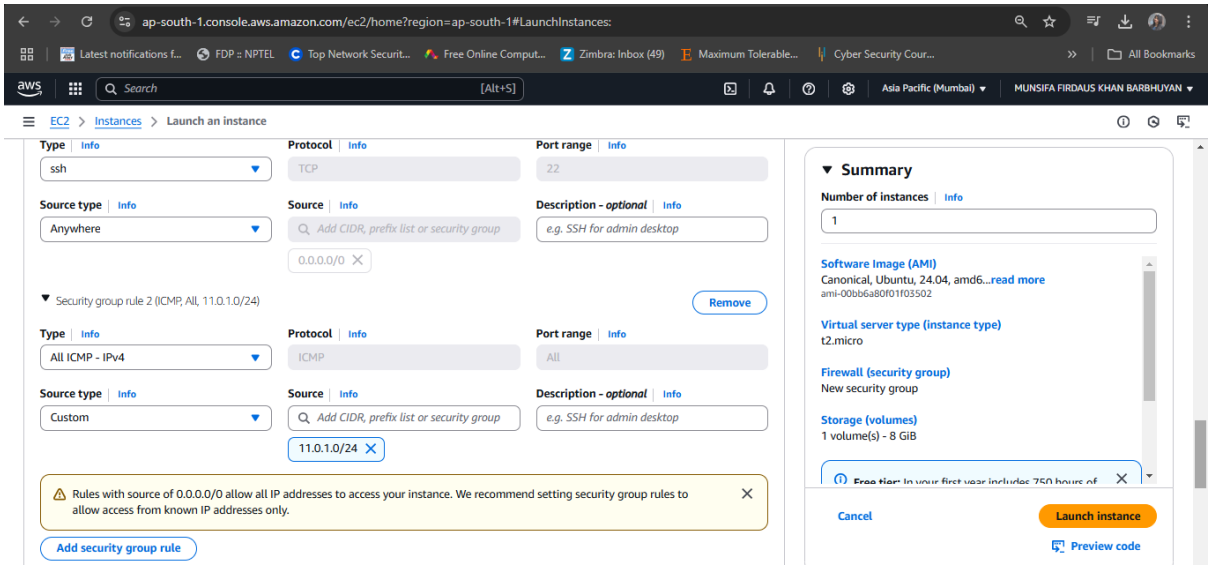


DISABLE AUTO-ASSIGN-PUBLIC IP.

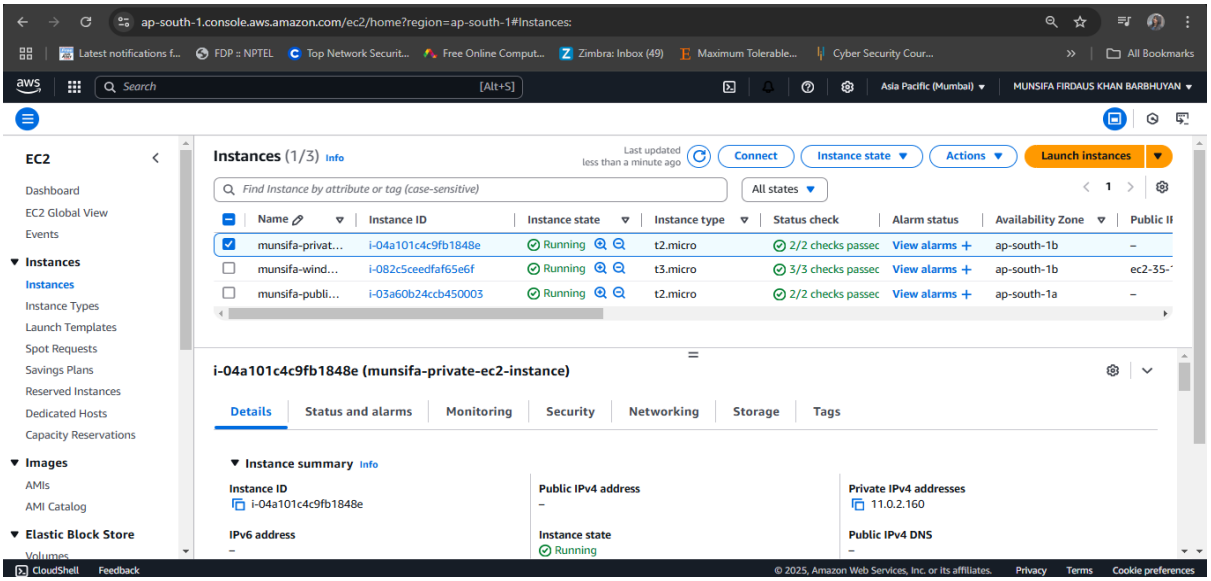




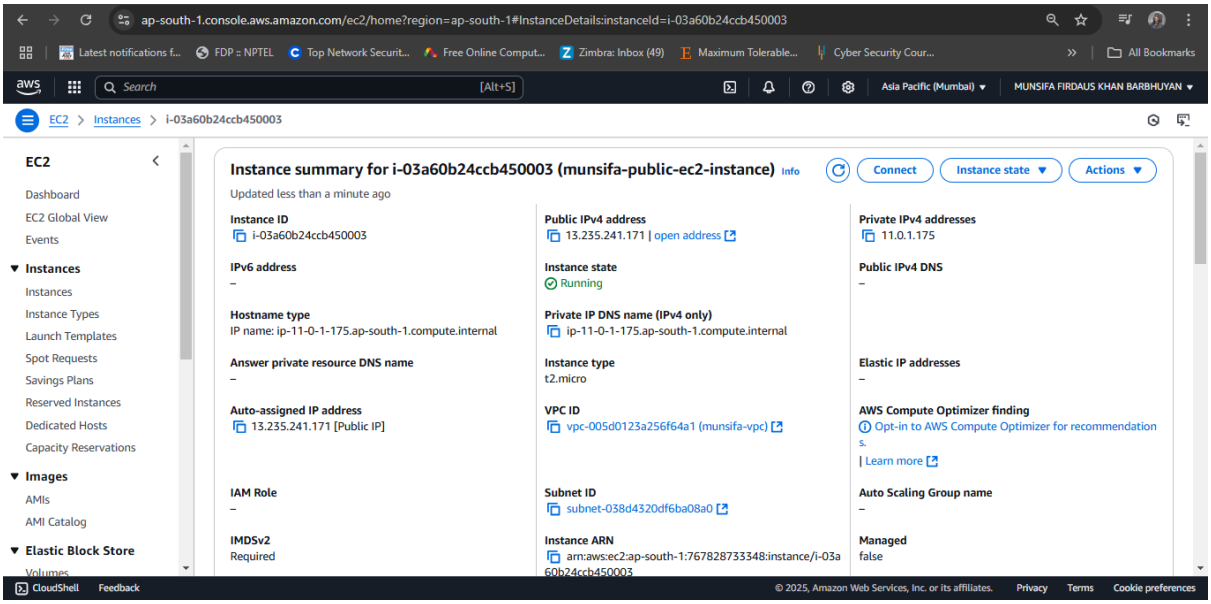
STEP 42: ADD SECURITY GROUP RULE>CHOOSE ALL ICMP-IPV4> SOURCE:11.0.1.0/24(PUBLIC SUBNET IP ADDRESS, COZ ONLY PUBLIC SUBNET CAN ACCESS THE PRIVATE SUBNET)>KEEP DEFAULT STORAGE>CLICK ON LAUNCH INSTANCE



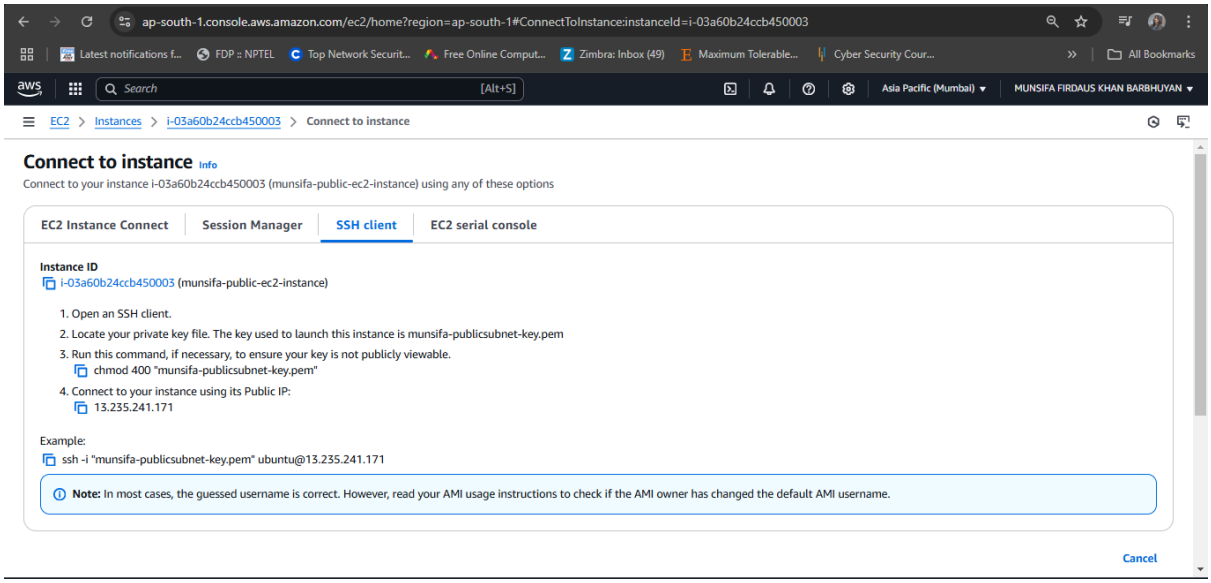
STEP 43: PRIVATE EC2 INSTANCE IS CREATED



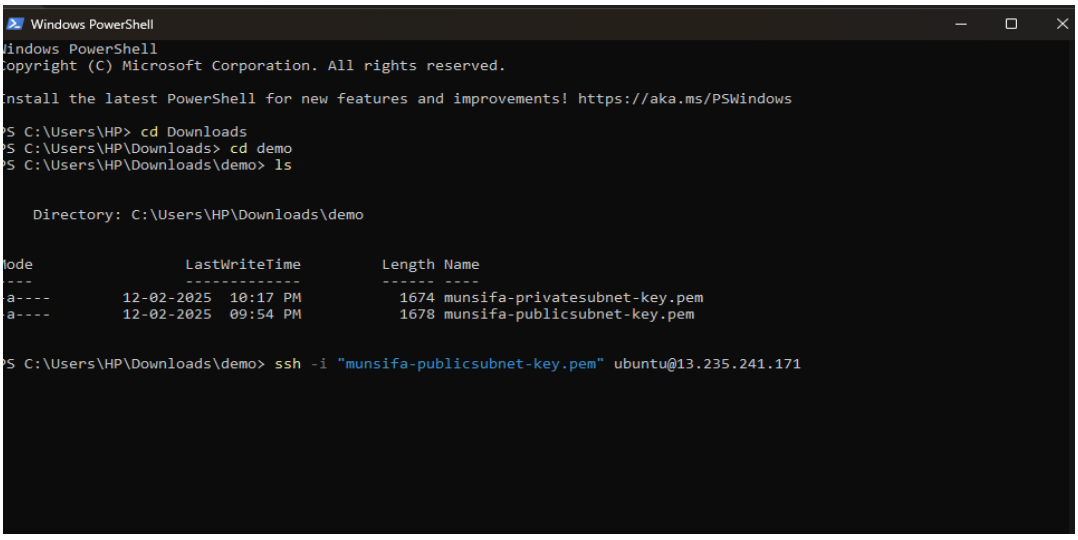
STEP 44: NOW CONNECT THE PUBLIC EC2 INSTANCE.CLICK ON THE PUBLIC INSTANCE> CLICK ON CONNECT



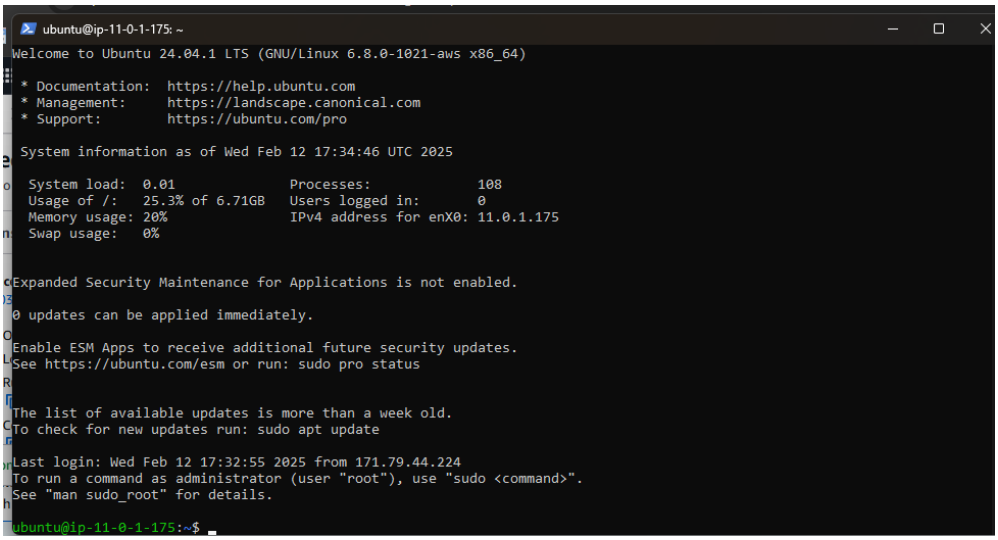
STEP 45: CLICK ON SSH CLIENT> COPY THE EXAMPLE AND PASTE IN TERMINAL TO CONNECT



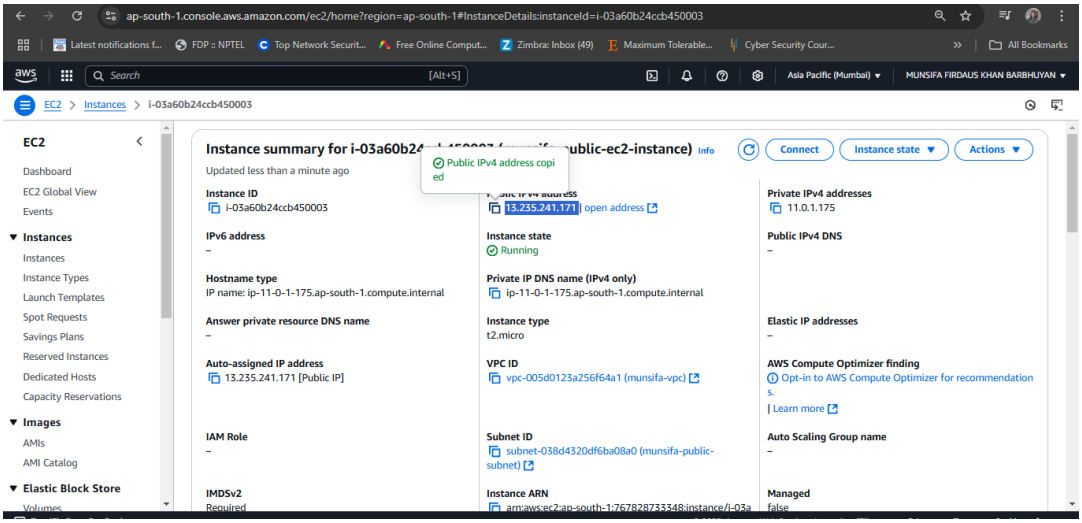
STEP 46: OPEN THE TERMINAL IN YOUR LAPTOP AND TYPE THE FOLLOWING COMMAND> NOW PASTE THE COMMAND HERE FROM THE SSH CLIENT

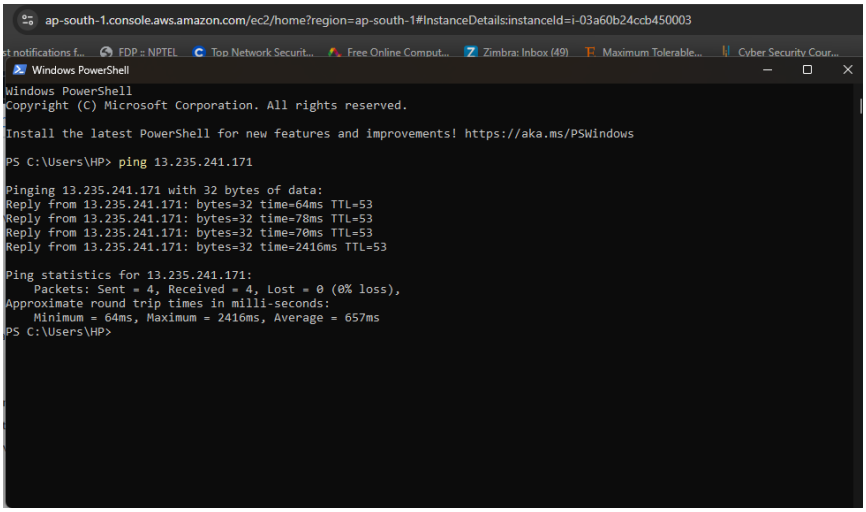


STEP 47: PUBLIC EC2 INSTANCE IS CONNECTED



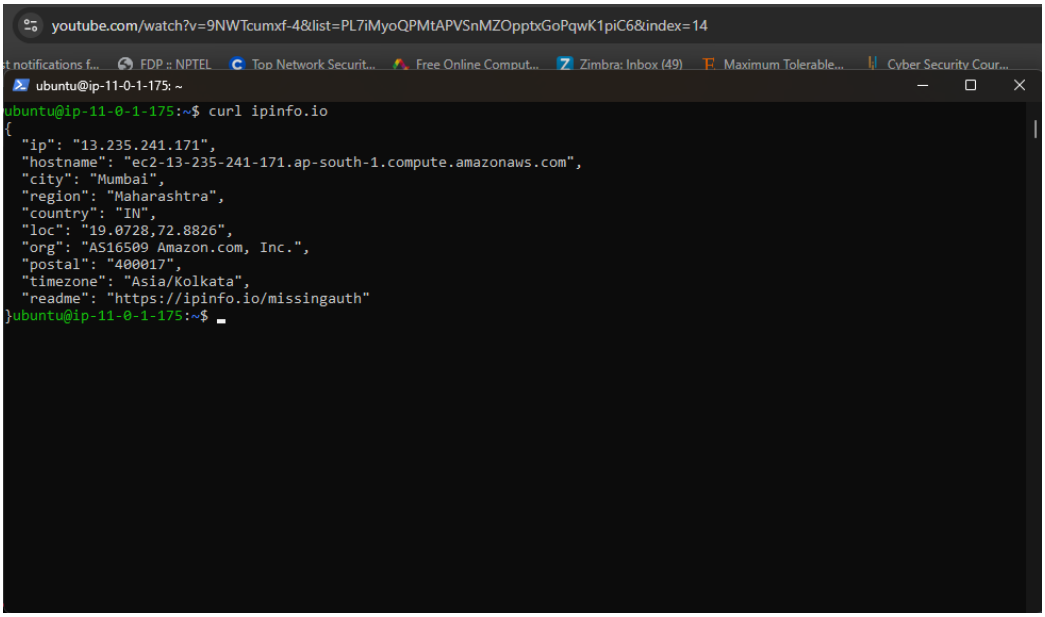
STEP 48: NOW PING THE PUBLIC EC2 INSTANCE WITH THE HELP OF YOUR LOCAL TERMINAL. GO TO INSTANCE AND COPY ITS IP ADDRESS AND TYPE THE FOLLOWING COMMAND. (IT MEANS FROM OUTSIDE AWS, WE ARE ABLE TO CONNECT TO THE PUBLIC EC2 INSTANCE)





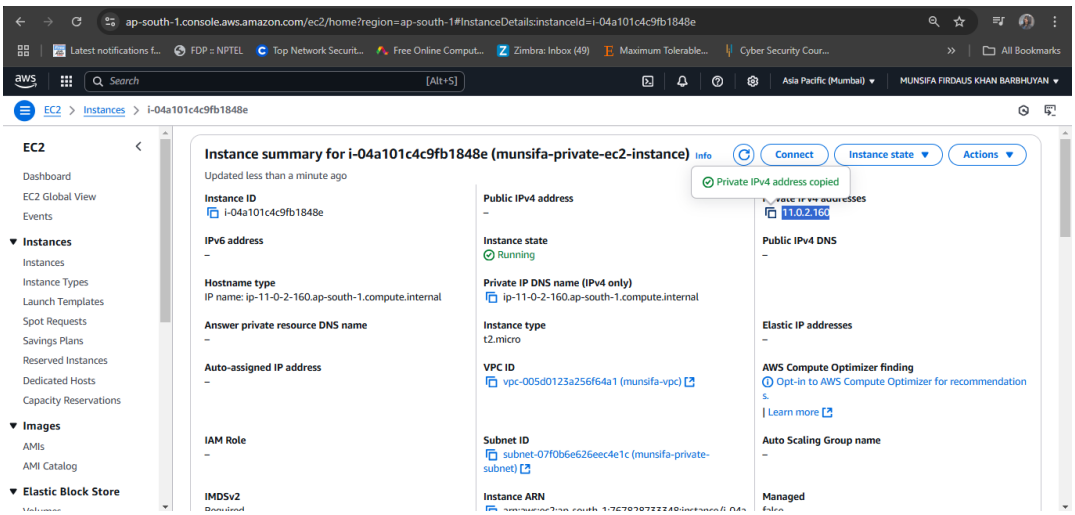
STEP 49: NOW NEXT FROM PUBLIC EC2 INSTANCE TERMINAL, TYPE THE FOLLOWING COMMAND TO ACCESS RESOURCE FROM OUTSIDE THE AWS

NOTE: curl ipinfo.io: It is a command used in the terminal to retrieve information about your current IP address using the "curl" tool and the "ipinfo.io" web service; essentially, it fetches details like your location, ISP, and public IP address by sending a request to the IPinfo.io API via the curl command.



It is seen that public EC2 instance is accessible from outside world as well as it can also access outside resources. But we know that, private instance that we have created under private subnet is not accessible from outside world but it can be accessed by the public EC2 instance only.

STEP 50: PING PRIVATE EC2 INSTANCE FROM THE LOCAL TERMINAL. COPY THE PRIVATE IP ADDRESS OF THE INSTANCE AND TYPE (PING 11.0.2.160) IN THE LOCAL TERMINAL.



```
PS C:\Users\HP> cd Downloads
PS C:\Users\HP\Downloads> cd demo
PS C:\Users\HP\Downloads\demo> ls


    Directory: C:\Users\HP\Downloads\demo


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12-02-2025  10:17 PM           1674 munsifa-privatesubnet-key.pem
-a----        12-02-2025  09:54 PM           1678 munsifa-publicsubnet-key.pem


PS C:\Users\HP\Downloads\demo> ping 11.0.2.160

Pinging 11.0.2.160 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 11.0.2.160:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PS C:\Users\HP\Downloads\demo>
```

ITS SHOWING REQUEST TIMED OUT. i.e., IT IS UNABLE TO CONNECT TO THE PRIVATE EC2 INSTANCE FROM THE LOCAL TERMINAL. NOW LET'S TRY TO CONNECT IT FROM THE PUBLIC EC2 INSTANCE TERMINAL.

STEP 50: TYPE THE FOLLOWING COMMAND TO CONNECT TO THE PRIVATE EC2 INSTANCE (PING 11.0.2.160). CONNECTION ESTABLISHED AS SHOWN BELOW.

```
ubuntu@ip-11-0-1-175: ~
ubuntu@ip-11-0-1-175:~$ curl ipinfo.io
{
  "ip": "13.235.241.171",
  "hostname": "ec2-13-235-241-171.ap-south-1.compute.amazonaws.com",
  "city": "Mumbai",
  "region": "Maharashtra",
  "country": "IN",
  "loc": "19.0728,72.8826",
  "org": "AS16509 Amazon.com, Inc.",
  "postal": "400017",
  "timezone": "Asia/Kolkata",
  "readme": "https://ipinfo.io/missingauth"
}ubuntu@ip-11-0-1-175:~$ ping 11.0.2.160
PING 11.0.2.160 (11.0.2.160) 56(84) bytes of data.
64 bytes from 11.0.2.160: icmp_seq=1 ttl=64 time=1.85 ms
64 bytes from 11.0.2.160: icmp_seq=2 ttl=64 time=0.955 ms
64 bytes from 11.0.2.160: icmp_seq=3 ttl=64 time=0.987 ms
64 bytes from 11.0.2.160: icmp_seq=4 ttl=64 time=1.11 ms
64 bytes from 11.0.2.160: icmp_seq=5 ttl=64 time=1.11 ms
64 bytes from 11.0.2.160: icmp_seq=6 ttl=64 time=0.840 ms
64 bytes from 11.0.2.160: icmp_seq=7 ttl=64 time=0.987 ms
64 bytes from 11.0.2.160: icmp_seq=8 ttl=64 time=1.02 ms
64 bytes from 11.0.2.160: icmp_seq=9 ttl=64 time=1.20 ms
64 bytes from 11.0.2.160: icmp_seq=10 ttl=64 time=1.09 ms
64 bytes from 11.0.2.160: icmp_seq=11 ttl=64 time=0.748 ms
64 bytes from 11.0.2.160: icmp_seq=12 ttl=64 time=1.07 ms
64 bytes from 11.0.2.160: icmp_seq=13 ttl=64 time=1.23 ms
64 bytes from 11.0.2.160: icmp_seq=14 ttl=64 time=0.984 ms
64 bytes from 11.0.2.160: icmp_seq=15 ttl=64 time=0.927 ms
```

RELEASE ALL THE RESOURCES THAT ARE USED IN THE EXECUTION OF THE LAB.

1. **TERMINATE THE EC2 INSTANCES.**
2. **DELETE ROUTE TABLE:**
   a. **ROUTE TABLE>EDIT ROUTE>REMOVE NAT GATEWAY (PRIVATE)**
   b. **ROUTE TABLE>EDIT ROUTE>REMOVE INTERNET GATEWAY (PUBLIC)**
   c. **ROUTE TABLE>SUBNET ASSOCIATIONS>EDIT SUBNET ASSOCIATIONS>DESELECT>SAVE CHANGES**
   d. **SELECT BOTH THE ROUTE TABLES>ACTIONS>DELETE ROUTE TABLES.**
3. **DELETE SUBNETS:**
   a. **SUBNETS> SELECT BOTH THE SUBNET ACTIONS>ACTIONS> DELETE SUBNET**
4. **DELETE NAT GATEWAY:**
   a. **NAT GATEWAY>SELECT THE NAT GATEWAY>ACTIONS> DELETE NAT GATEWAY**
5. **DELETE VPC**
   a. **VPC>SELECT THE VPC CREATED>ACTIONS>DELETE**
6. **RELEASE ELASTIC IP**
   a. **ELASTIC IP>SELECT THE ELASTIC IP>ACTIONS>RELEASE THE ELASTIC IP**