

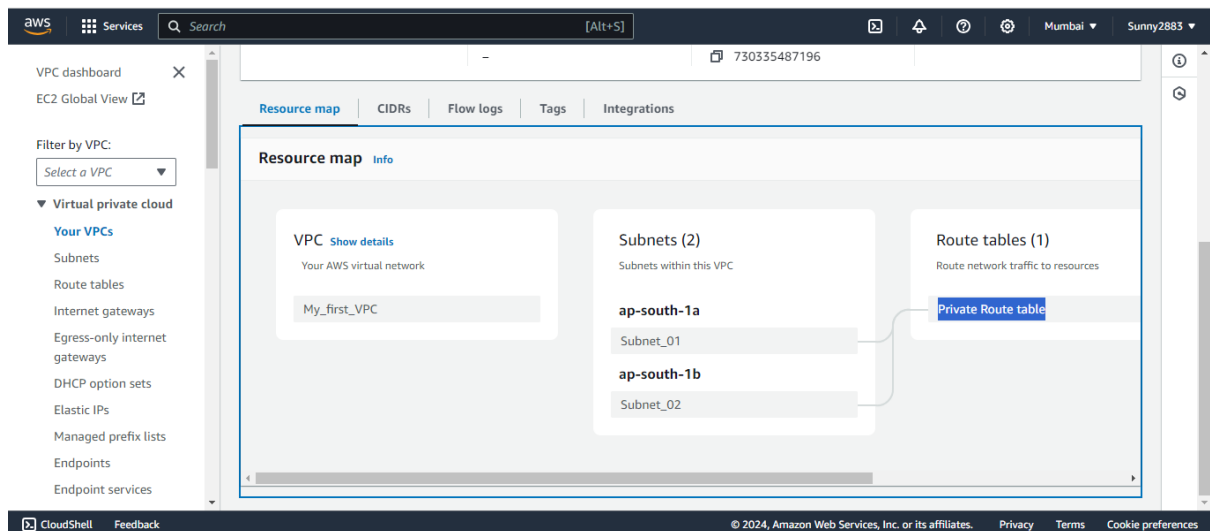
Assignment VPC

Task:2

Subnet Configuration:

Configure one subnet as a public subnet and the other as a private subnet.

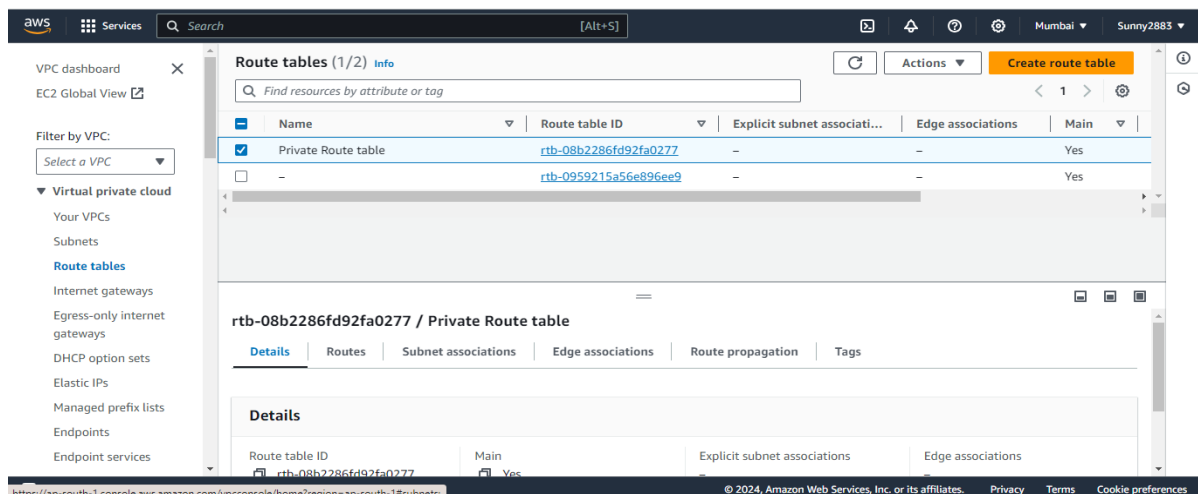
In Task One, I successfully established "My_first_VPC," a Virtual Private Cloud on AWS. Within this VPC, I meticulously crafted two subnets – Subnet_01 and Subnet_02. Both subnets are seamlessly integrated with a dedicated route table named "Private Route table."



Private Route table

Let's Create one more route table and give the route table and attach with subnet_02 to Make it public .

Step1:Create a route table.



Step2: Go to the AWS VPC dashboard. On the left sidebar, click on "Route Tables."

Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

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

VPC
The VPC to use for this route table.

Tags
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="public Route table"/>	<input type="button" value="Remove"/>

Step2: Click on the "Create Route Table" button. Enter a name for your route table.

Subnets (1/5) [Info](#)

	Name	Subnet ID	State	VPC	IPv4 CIDR
<input checked="" type="checkbox"/>	Subnet_02	subnet-07fcf424c6757106f	Available	vpc-0656d6ae2918c1d0a My...	192.168.2.0/24
<input type="checkbox"/>	-	subnet-0e6cdf2a06c9753a9	Available	vpc-06988e5057435e138	172.31.0.0/24
<input type="checkbox"/>	Subnet_01	subnet-013c0000f10df46e8	Available	vpc-0656d6ae2918c1d0a My...	192.168.0.0/24
<input type="checkbox"/>	-	subnet-0c714c65c4325ac96	Available	vpc-06988e5057435e138	172.31.0.0/24

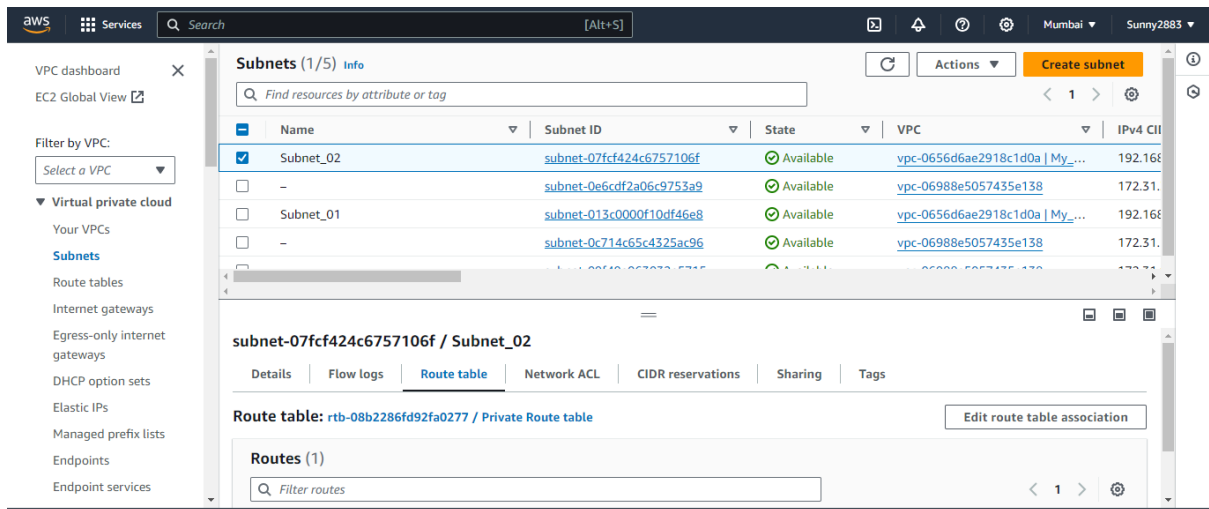
subnet-07fcf424c6757106f / Subnet_02

[Details](#) [Flow logs](#) [Route table](#) [Network ACL](#) [CIDR reservations](#) [Sharing](#) [Tags](#)

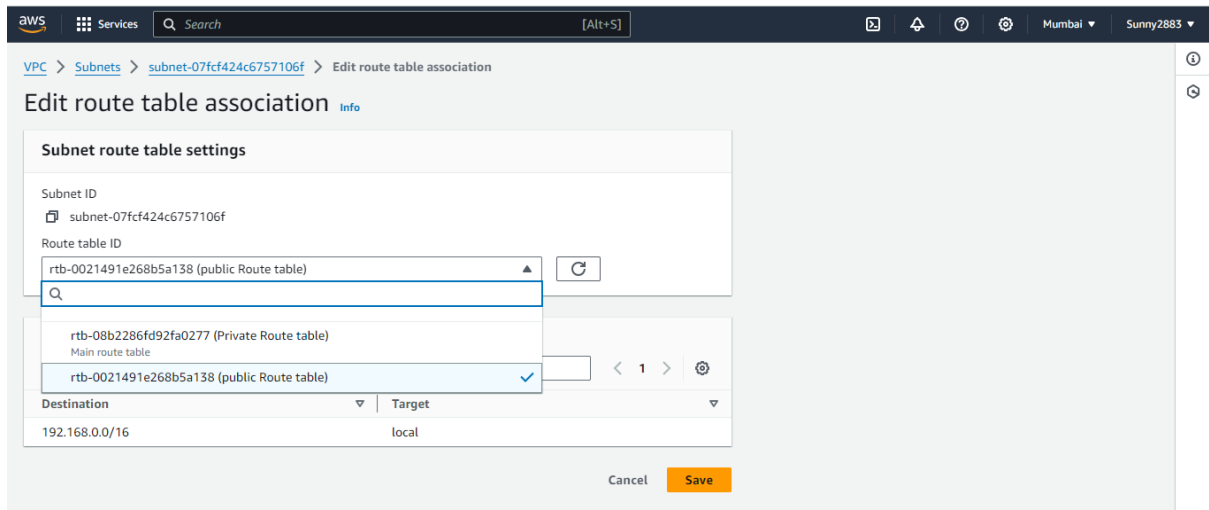
Details

Subnet ID <input type="text" value="subnet-07fcf424c6757106f"/>	Subnet ARN <input type="text" value="arn:aws:ec2:ap-south-1:730335487196:subnet/subnet-07fcf424c6757106f"/>	State <input checked="" type="checkbox"/> Available	IPv4 CIDR <input type="text" value="192.168.2.0/24"/>
--	--	--	--

Step3: Once the route table is created, select it from the list. In the "Routes" tab, click on "Edit routes."



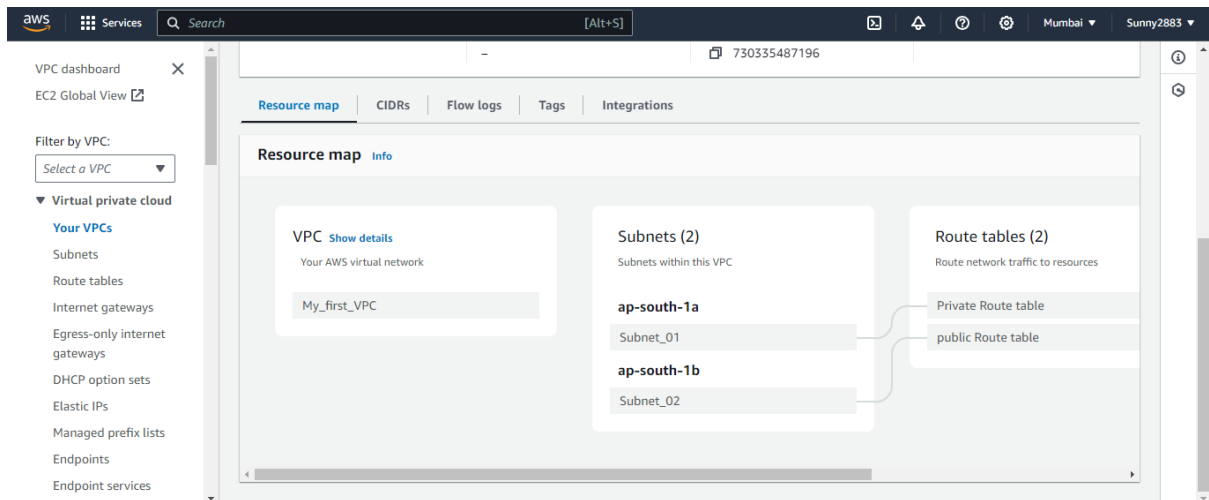
Step4: Change private route table to public route table and save the changes.



Now the both subnets are associated with two different route tables.

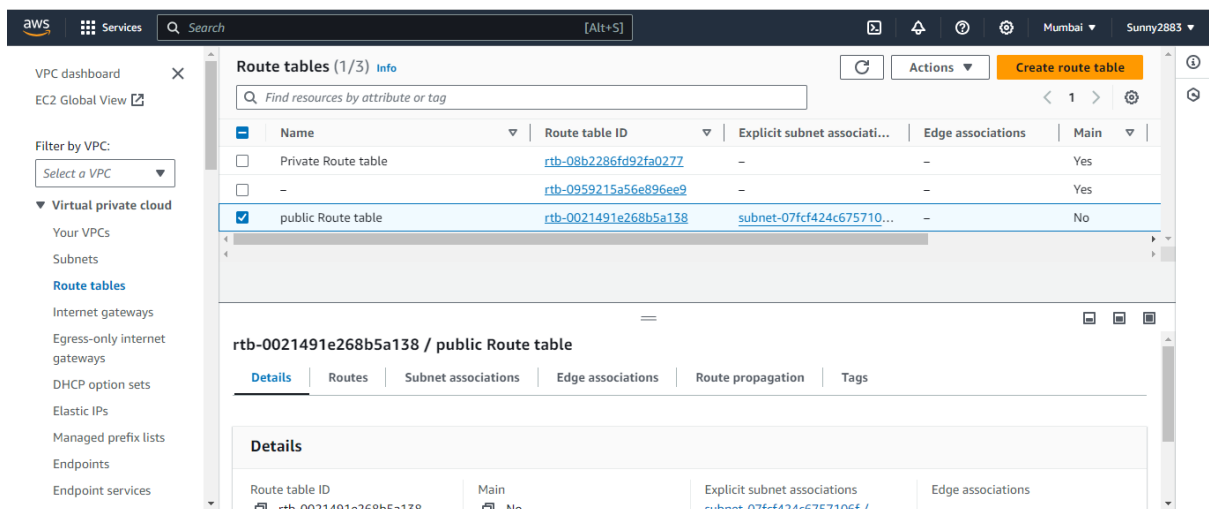
Subnet_01 → private route table. For (private Subnet)

Subnet_02 → public route table. For (Public Subnet)

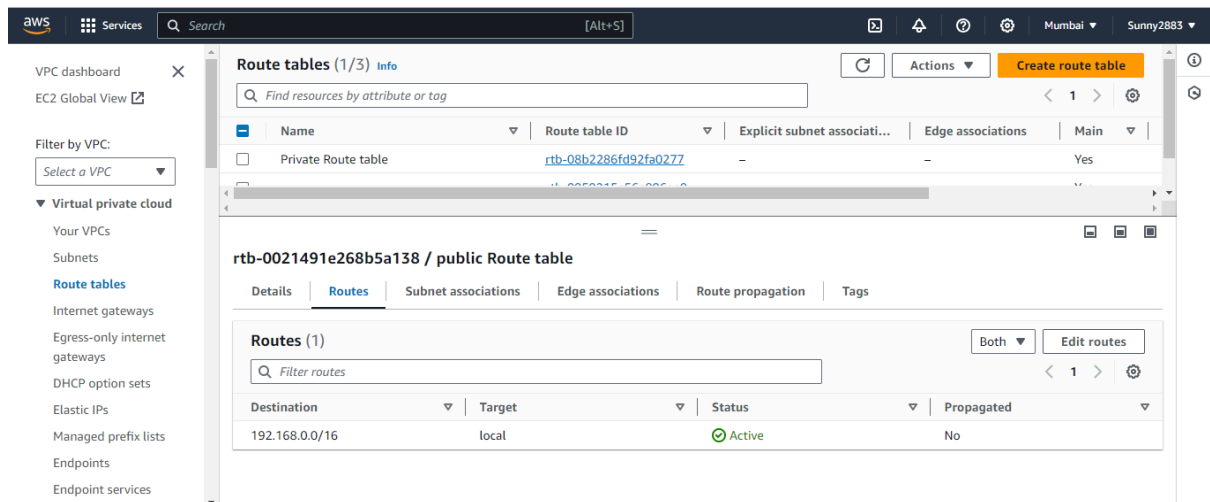


Now make the changes in to public route table.

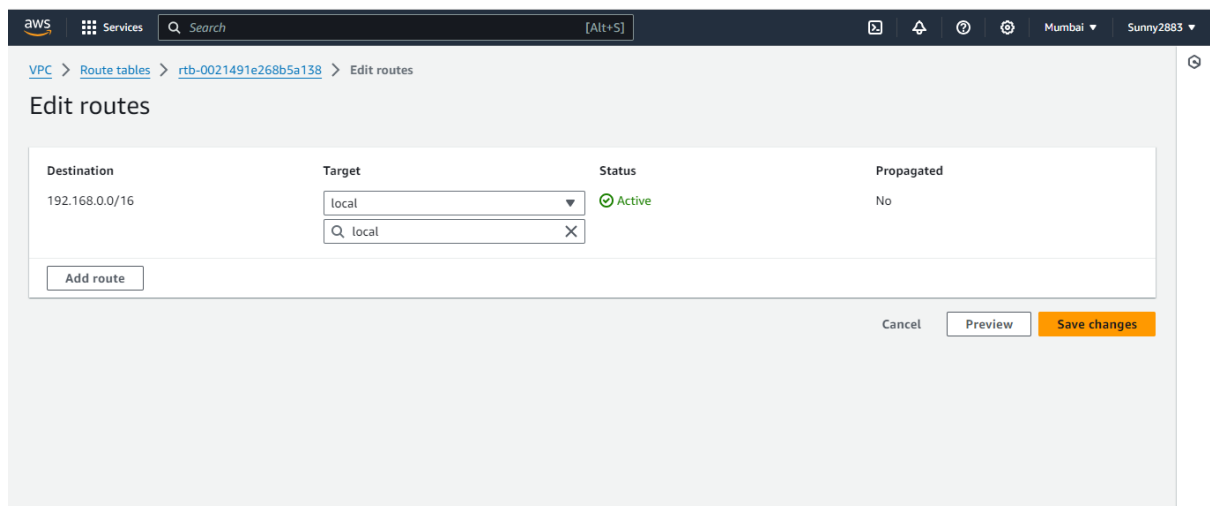
Step1: Goto Route Table and select public route table ;



Step2: In Public Route table Goto route and select edit route.



Step3: Select add route.

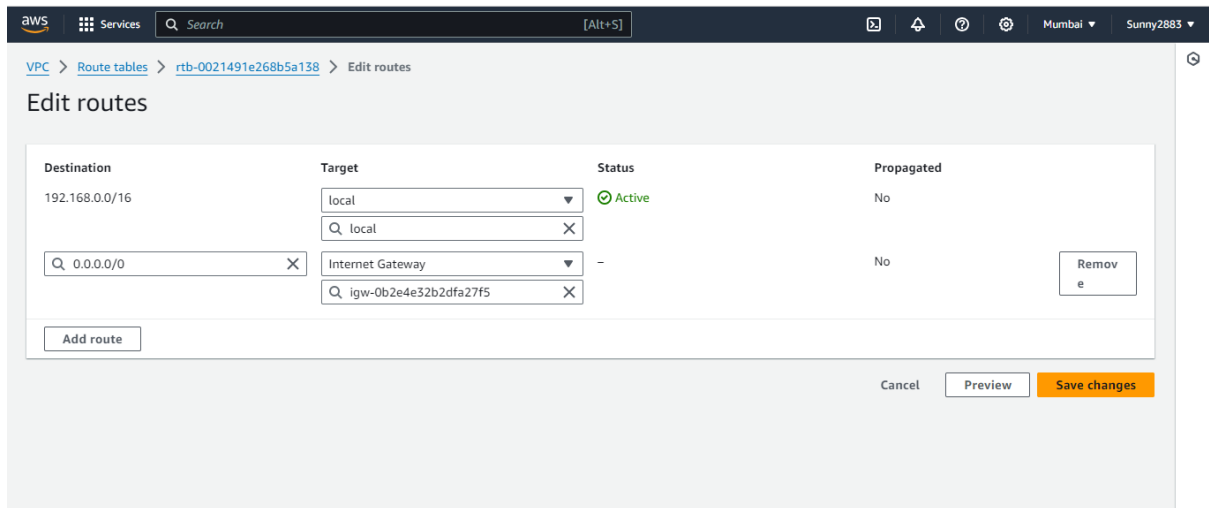


Step4: Make necessary changes to make it public .

Add 0.0.0.0/0 in Destination and select Internet Gateway and choose (igw-0b2e4e32b2dfa27f5 my_first_internet_gateway) and save the changes.

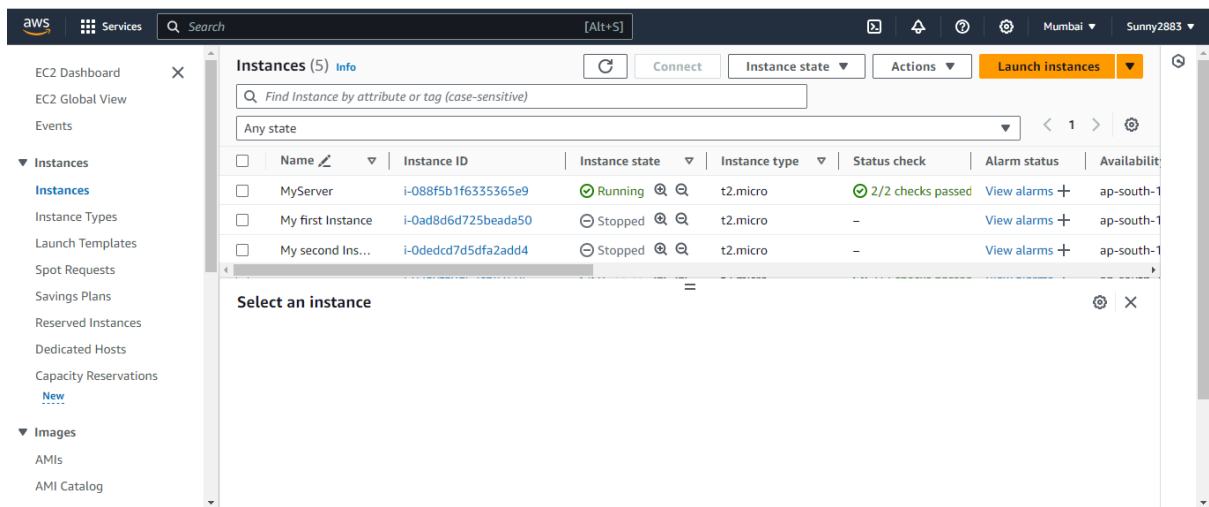
Destination--- 0.0.0.0/0

Target—Internet gateway (igw-0b2e4e32b2dfa27f5 my_first_internet_gateway)

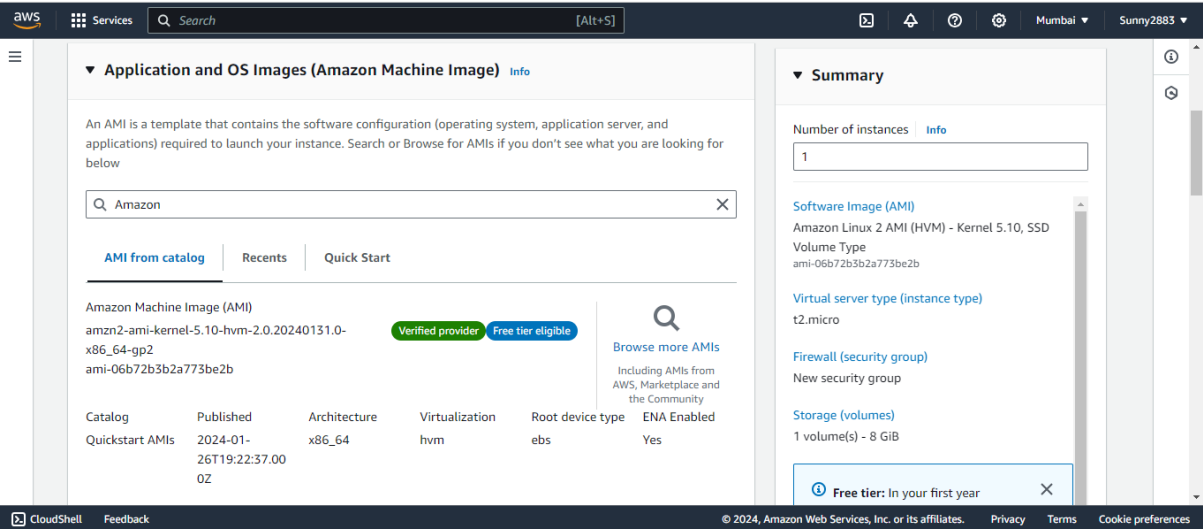


2. Launch an EC2 instance in each subnet. The EC2 instance in the public subnet should be reachable from the Internet.

Step1: Go to the EC2 dashboard. Click on the "Launch Instance" button.

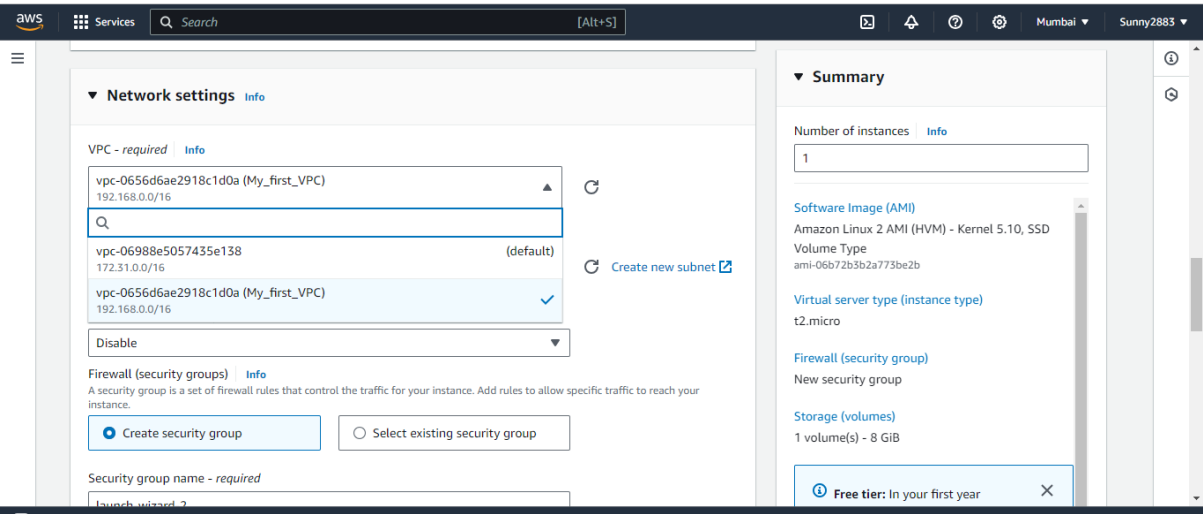


Step2: Choose an Amazon Machine Image (AMI) and select the instance type.



Step3: Configure instance details:

VPC: [vpc-0656d6ae2918c1d0a](#) (My_first_VPC)



Step4: Select Subnet.

Subnet: Subnet_01.

Security group: Create Security Group.

Type: SSH

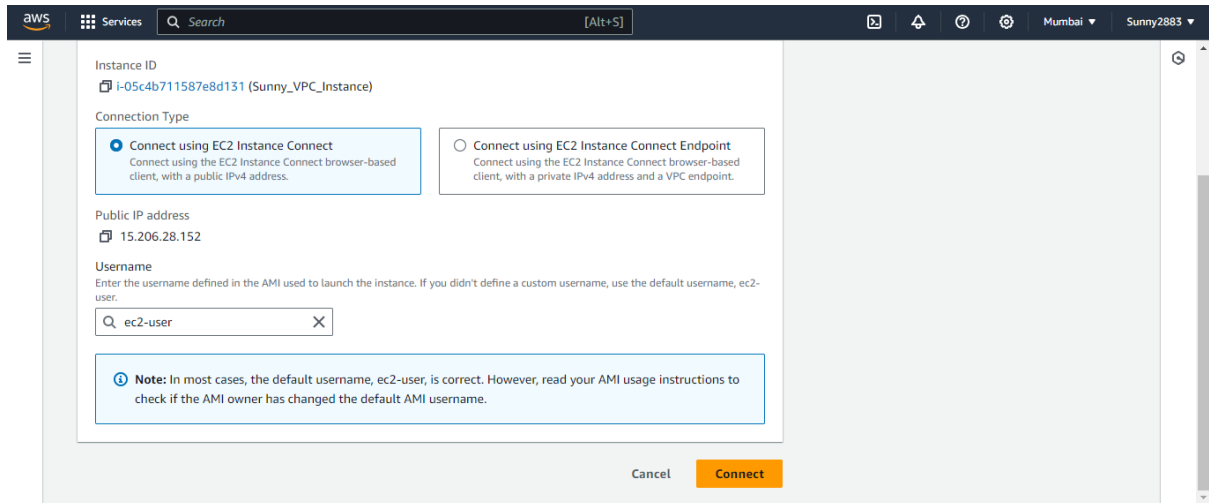
Source type: Anywhere

The screenshot shows the 'Create Security Group' page in the AWS Management Console. The 'Security group name' is 'launch-wizard-3'. The 'Description' is 'launch-wizard-3 created 2024-02-05T11:46:02.777Z'. Under 'Inbound Security Group Rules', a rule is added for 'ssh' on 'TCP' port '22' with 'Source type' set to 'Anywhere'. The 'Summary' panel on the right shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2 AMI (HVM)', 'Virtual server type (instance type)' as 't2.micro', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A 'Free tier' banner is visible at the bottom right.

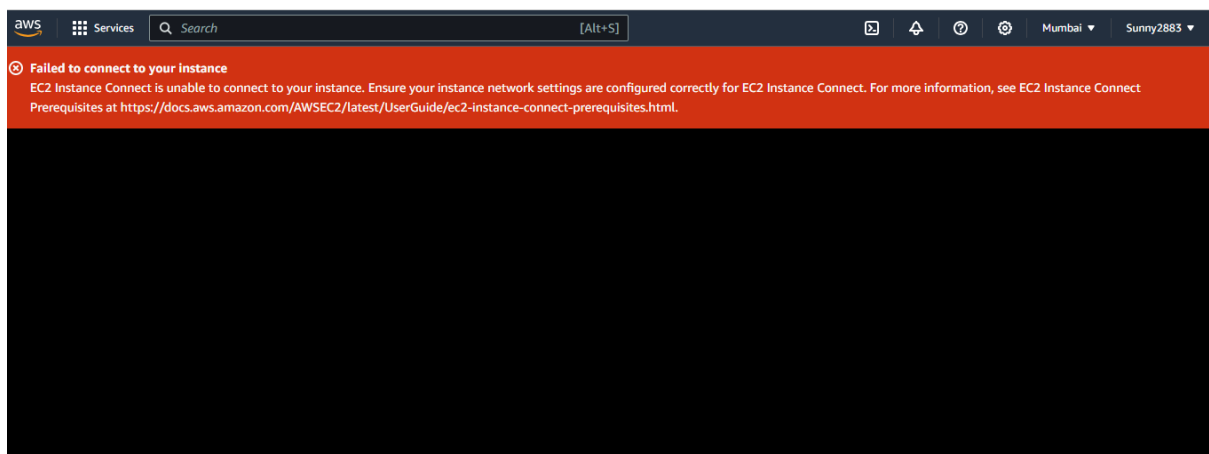
Step5: Review and launch the instance.

The screenshot shows the 'Configure storage' page in the AWS Management Console. The 'Root volume' is configured as '8 GiB' with 'gp2' storage type. A 'Free tier' banner is visible. The 'Advanced details' section is expanded, showing '0 x File systems'. The 'Summary' panel on the right shows the same configuration as the previous step. At the bottom, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.

Step6: Let's try to Connect to the instance.



Step7: We are unable to connect to the instance.



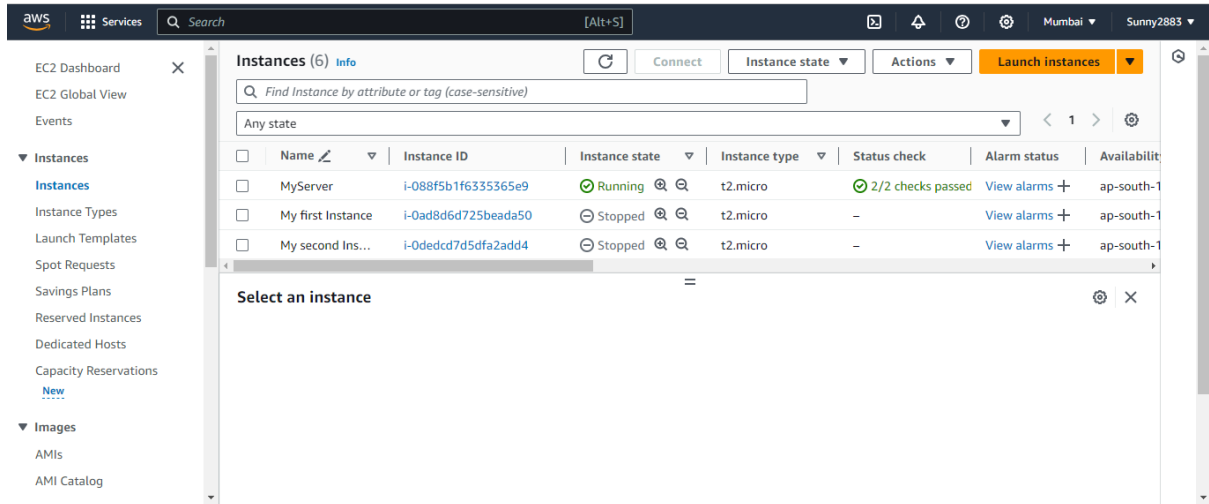
Instance link with private subnet.

Link:<https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-05c4b711587e8d131&osUser=ec2-user&sshPort=22#/>

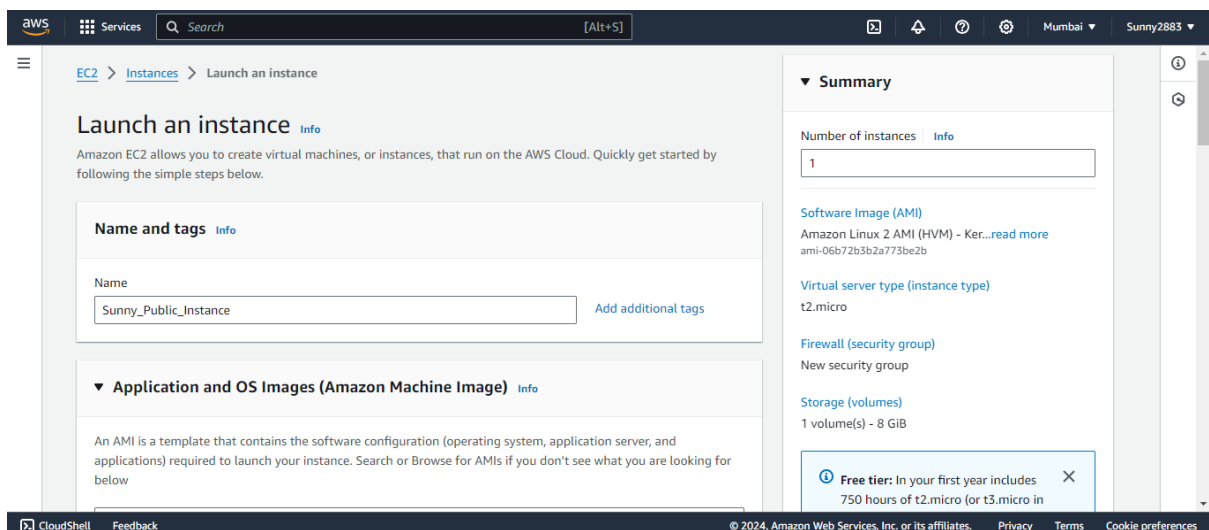
Public Subnet:

Create an instance with public subnet.

Step1: Go to the EC2 dashboard. Click on the "Launch Instance" button.



Step2: Choose an Amazon Machine Image (AMI) and select the instance type.



Step3: Configure instance details:

Select the "My_first_VPC" VPC.

Choose the "Subnet_02" public subnet.

Enable "Auto-assign Public IP" to allow the instance to be reachable from the internet.

VPC: [vpc-0656d6ae2918c1d0a](#) (My_first_VPC)

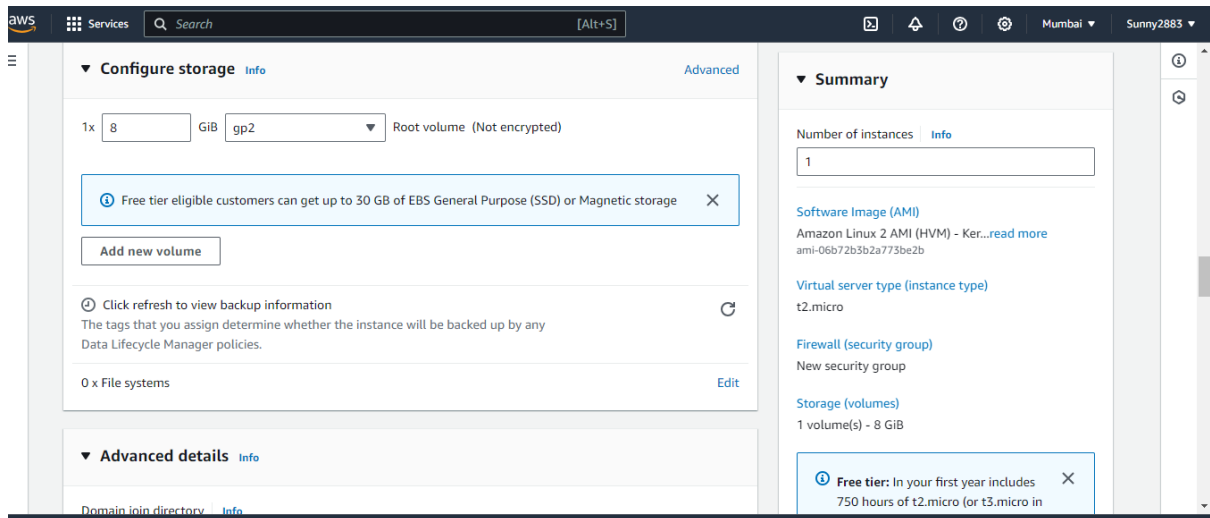
Subnet: Subnet_02

The screenshot shows the 'Network settings' tab in the AWS Management Console. The 'VPC' dropdown is set to 'vpc-0656d6ae2918c1d0a (My_first_VPC)'. The 'Subnet' dropdown is set to 'subnet-07fcf424c6757106f Subnet_02'. The 'Auto-assign public IP' dropdown is set to 'Enable'. The 'Firewall (security groups)' section shows a 'Create security group' button and a 'Select existing security group' button. The 'Security group name - required' field is 'launch-wizard-2'. The 'Summary' tab on the right shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2 AMI (HVM) - Ker...', 'Virtual server type (instance type)' as 't2.micro', and 'Firewall (security group)' as 'New security group'. A 'Free tier' banner at the bottom right indicates 750 hours of t2.micro or t3.micro in the first year.

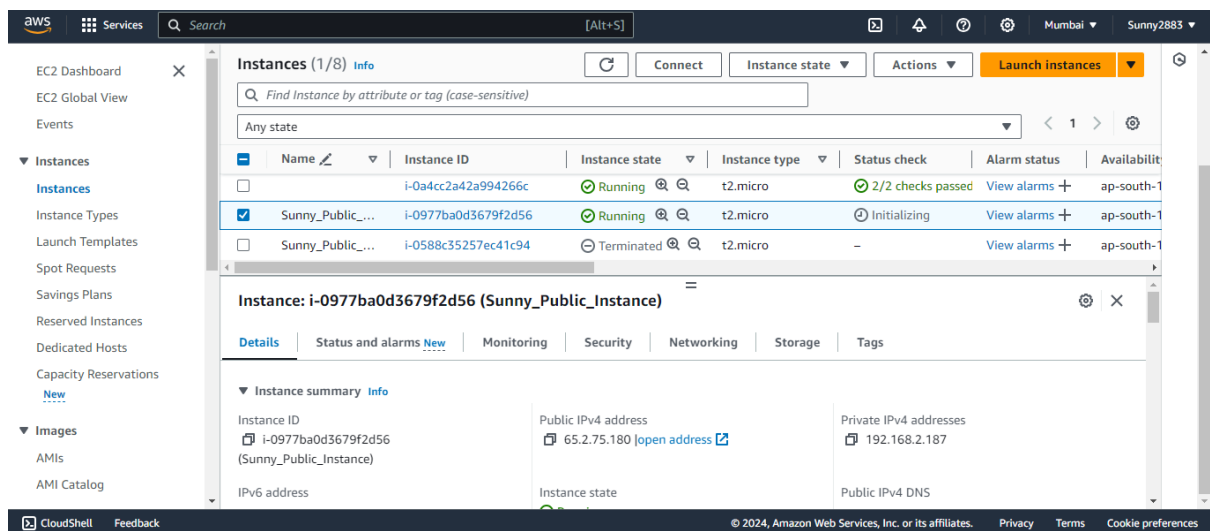
Step4: Add storage, configure tags, and define security groups.

The screenshot shows the 'Security groups' tab in the AWS Management Console. The 'Security group name - required' field is 'launch-wizard-2'. The 'Description - required' field is 'launch-wizard-2 created 2024-02-05T11:29:06.838Z'. The 'Inbound Security Group Rules' section shows a rule for 'ssh' (Type), 'TCP' (Protocol), and '22' (Port range). The 'Source type' is 'Anywhere' and the 'Source' is '0.0.0.0/0'. The 'Summary' tab on the right shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2 AMI (HVM) - Ker...', 'Virtual server type (instance type)' as 't2.micro', and 'Firewall (security group)' as 'New security group'. A 'Free tier' banner at the bottom right indicates 750 hours of t2.micro or t3.micro in the first year.

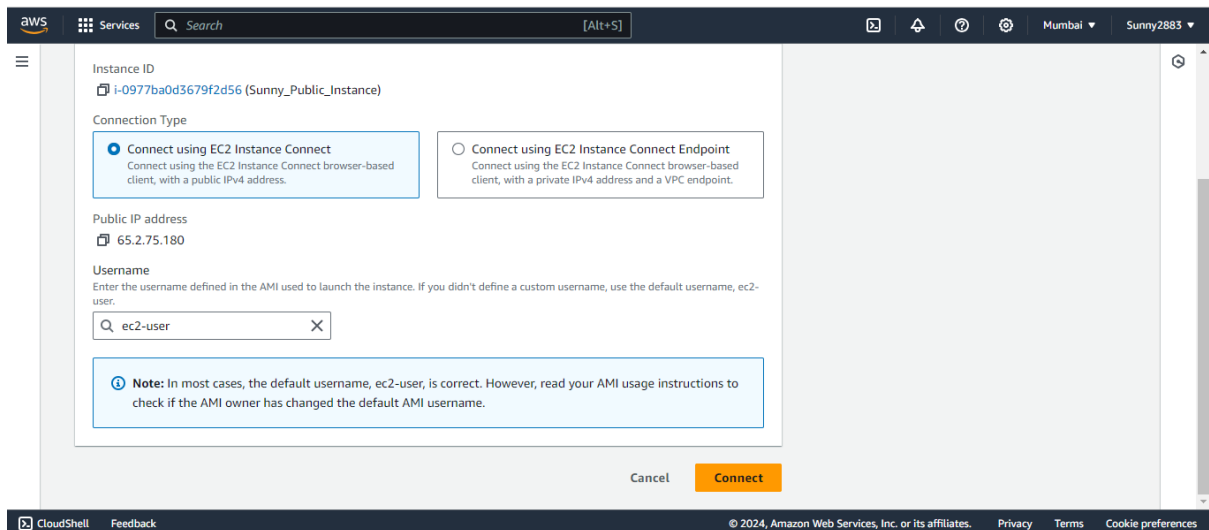
Step5: Review and launch the instance.



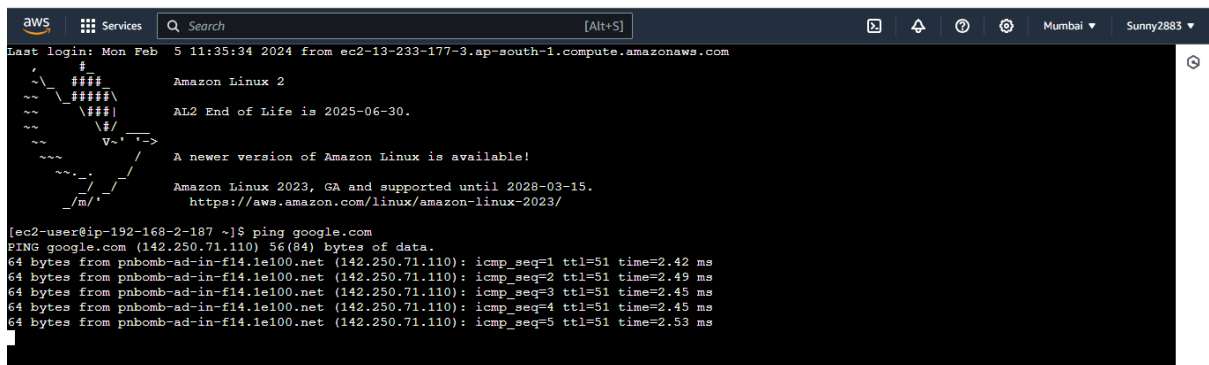
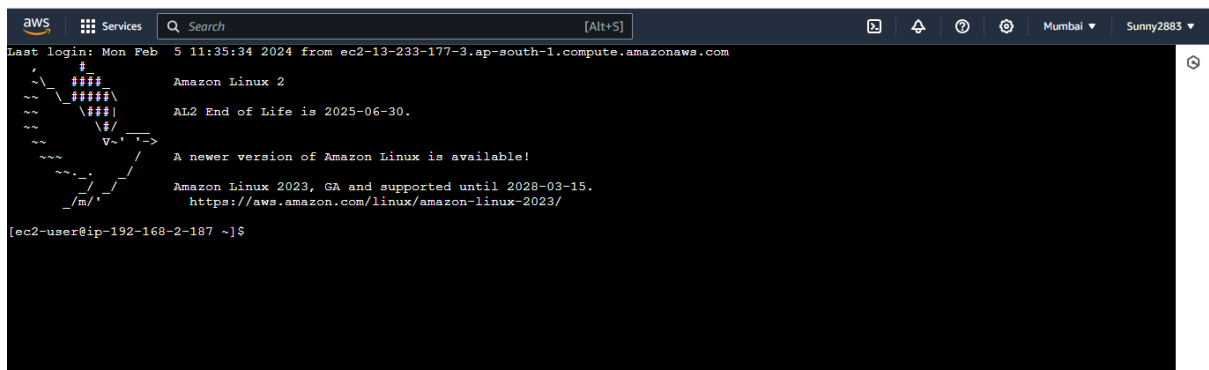
Step6: Once the instances are running, note the public IP of the EC2 instance in the public subnet.



Step7: Connect using EC2 instance connect.



Step8: From the public EC2 instance, run commands like ping or curl to verify internet connectivity.



Link of EC2 Instance with public Subnet

Link: <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0977ba0d3679f2d56&osUser=ec2-user&sshPort=22#/>

Ap-south-1b connected to internet gateway(my_first_internet_gateway)

