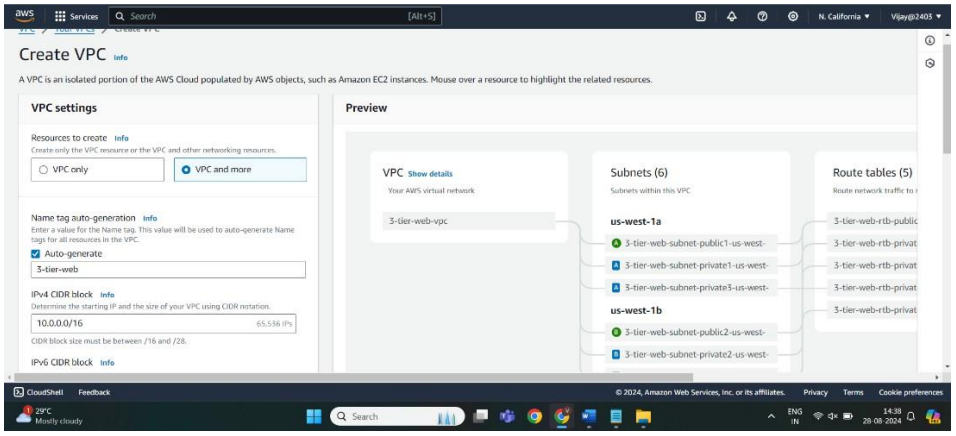
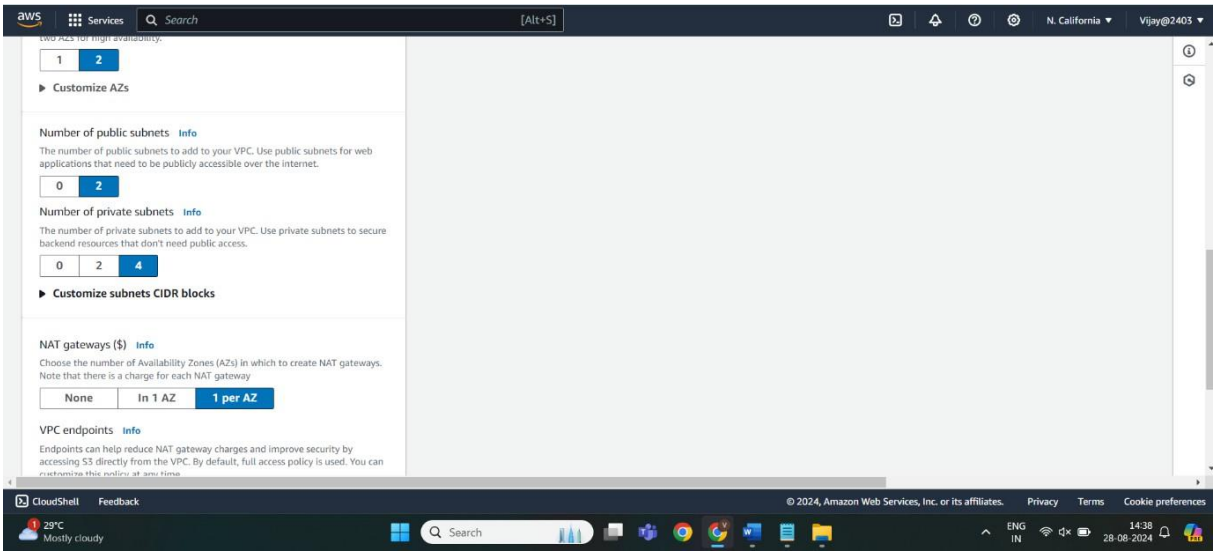
**PROJECT-1**

**3-TIER ARCHITECTURE**

Navigate to VPC > Create VPC > Choose "VPC and more" option. Set the IPv4 CIDR block to 10.0.0.0/16, select "No" for the IPv6 CIDR block, and leave the Tenancy option as "Default."

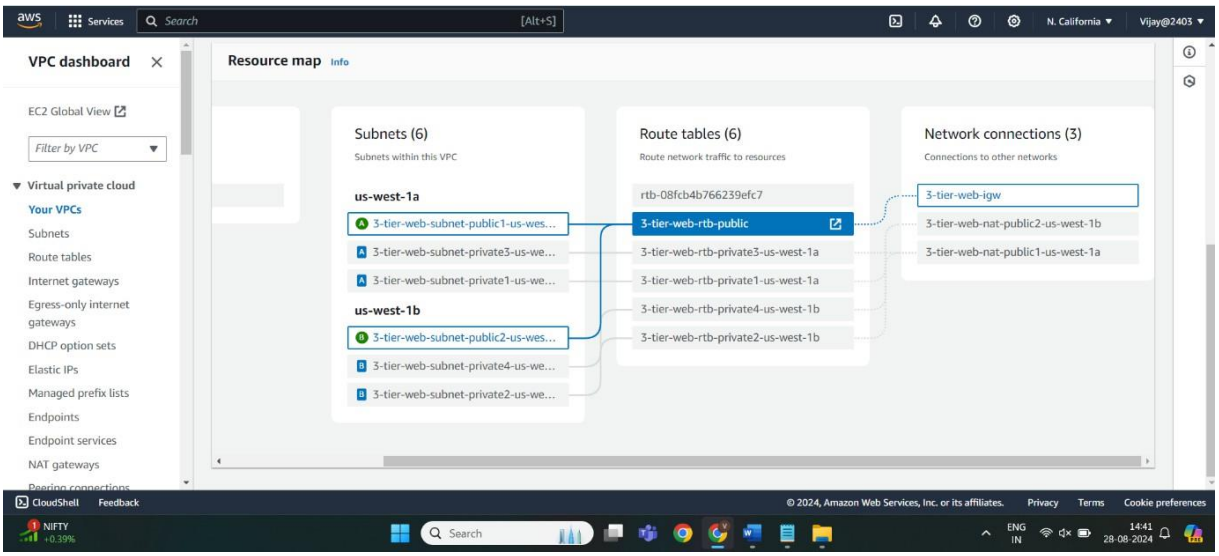


- Select ‘2’ Availability Zones.
- Set the First availability zone to us-west-1a.
- Set the Second availability zone to us-west-1b.
- Enter the public and private subnet CIDR information as specified.
- Choose 1 per AZ for the NAT gateway.
- Select None for VPC endpoints.
- Enable DNS hostnames and DNS resolution.

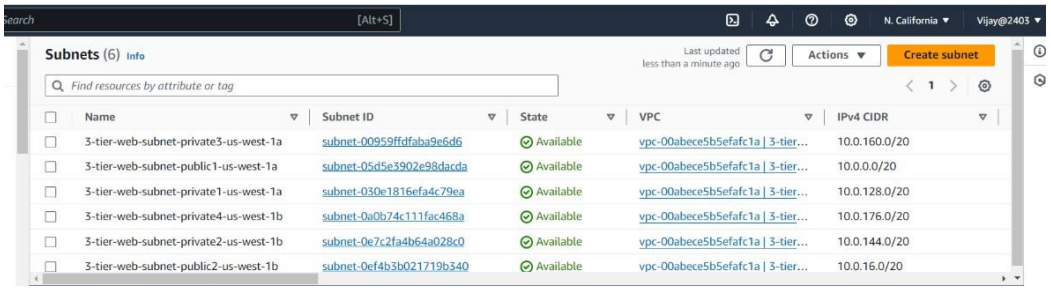


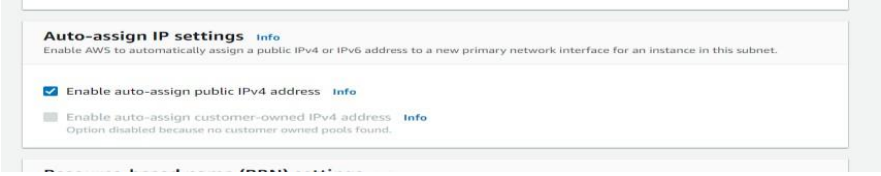

**- Click the Create VPC button.**

Once completed, click View VPC button. Now, we have a VPC, 2 public subnets, 4 private subnets, and 2 NAT gateways.

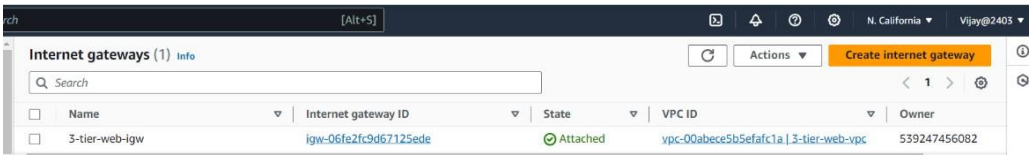



- Navigate to VPC > Subnets.
- Select one of the two public subnets.
- Click the Actions menu button.
- Choose Edit subnet settings.
- Enable Auto-assign public IPv4 address.
- Click the Save button.
- Repeat the process for the other public subnet.

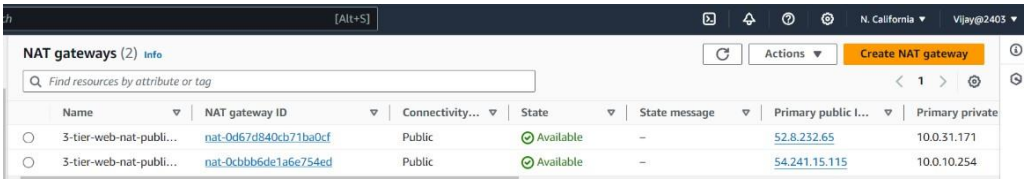


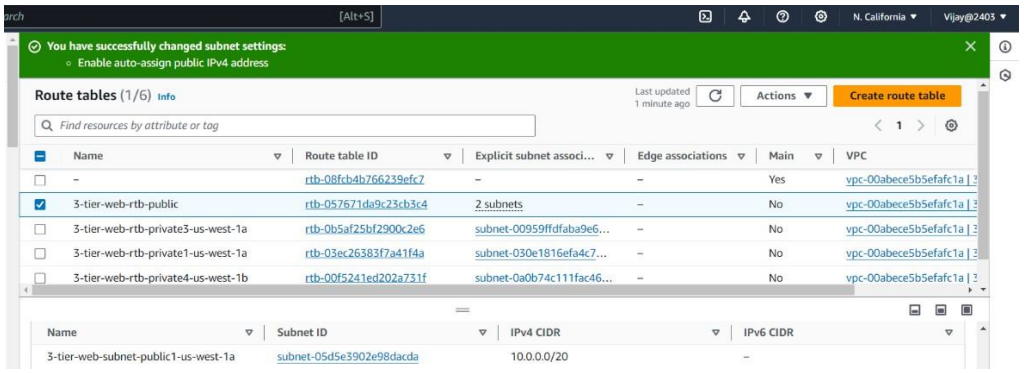

Navigate to Internet Gateway and confirm that Internet Gateway is attached to the VPC we just created.




Navigate to NAT gateways and confirm that two NAT gateways are created.



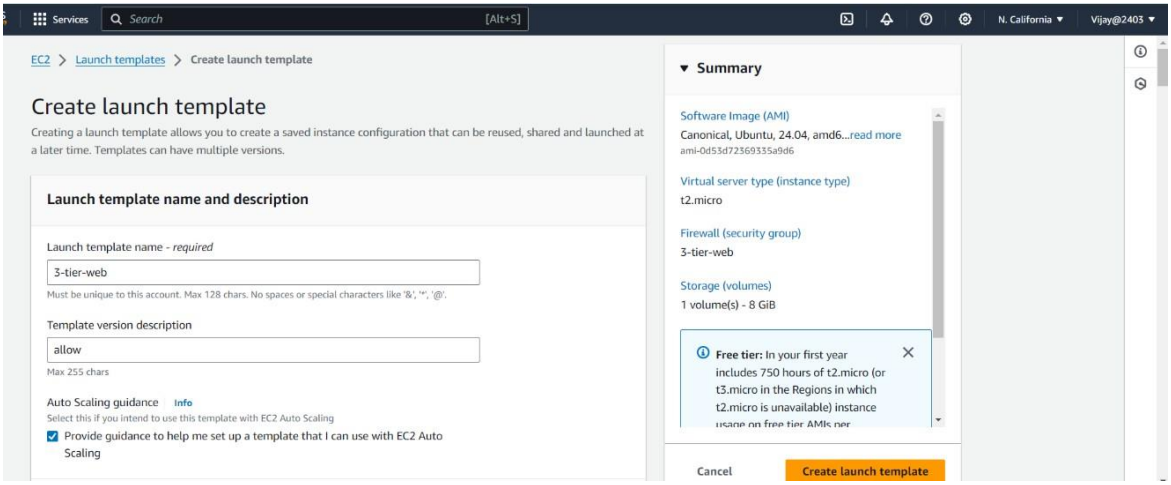

- Navigate to the public Route tables.
- Confirm that both public subnets are explicitly associated.




- Verify that each private route table is explicitly associated with its corresponding private subnet.

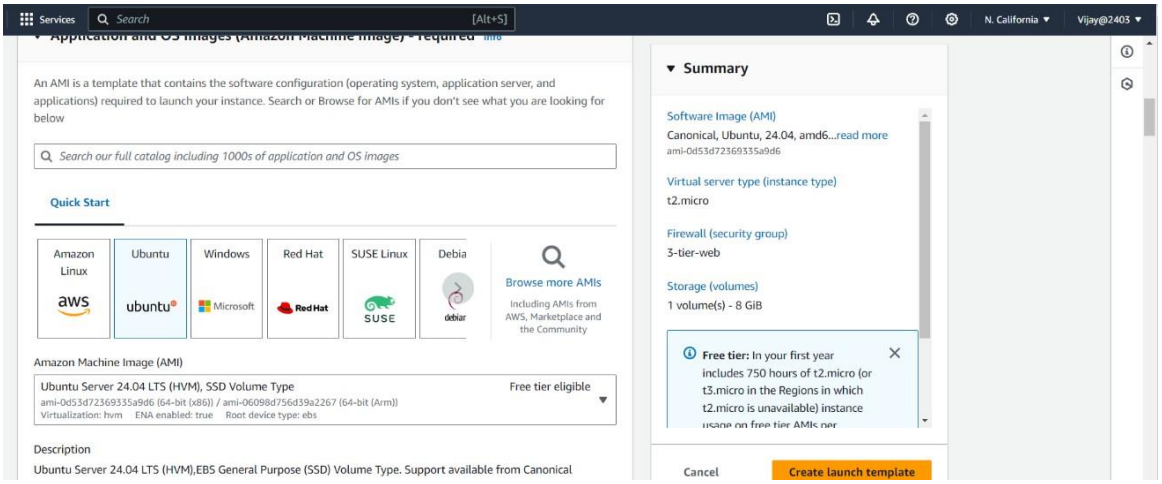
STEP-1: CREATE WEB TIER

- Navigate to EC2 > Launch Templates.
- Click the Create launch template button.
- Enter the launch template name and description.

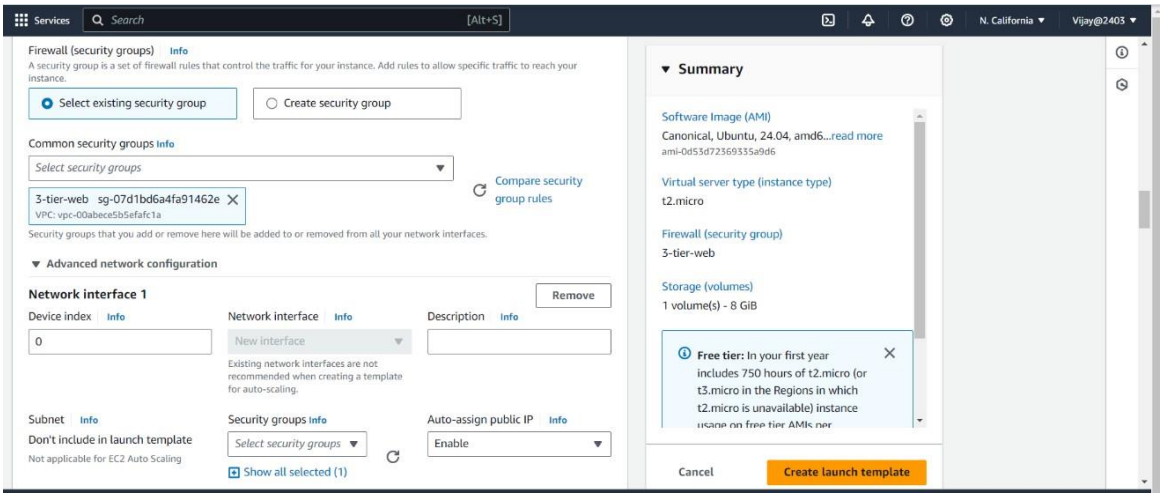


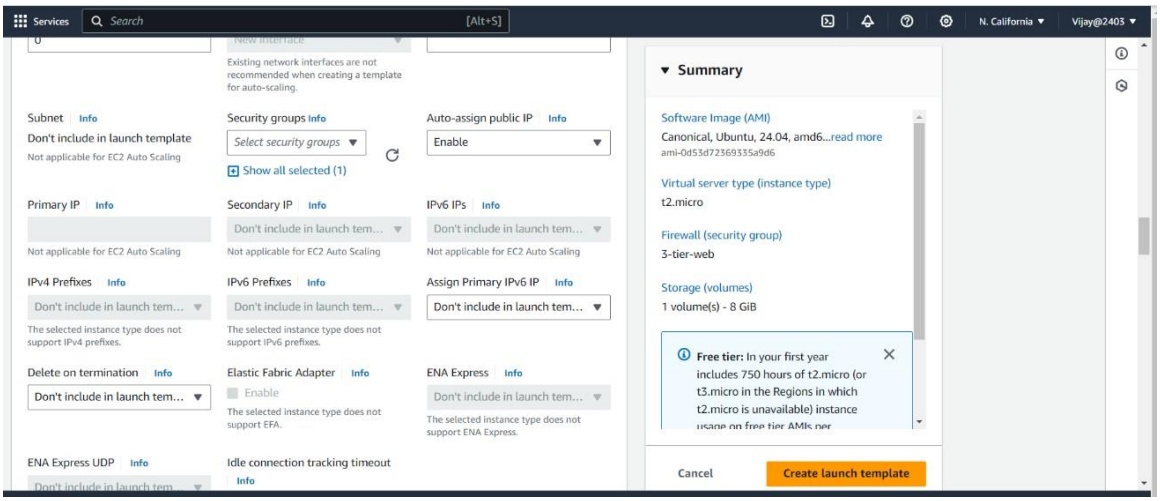

- Check the box for Auto Scaling guidance.

Select your AMI and the Instance type. I selected Ubuntu server 24.04 LTS and t2.micro Instance type.




- Select the Security Group and provide it with a name.
- Choose the VPC created in Part 1.
- Click the Add Security Group Rules button.
- Add rules to allow traffic on ports for ICMP, HTTP, and SSH.
- Under Advanced network configuration, enable Auto-assign public IP.

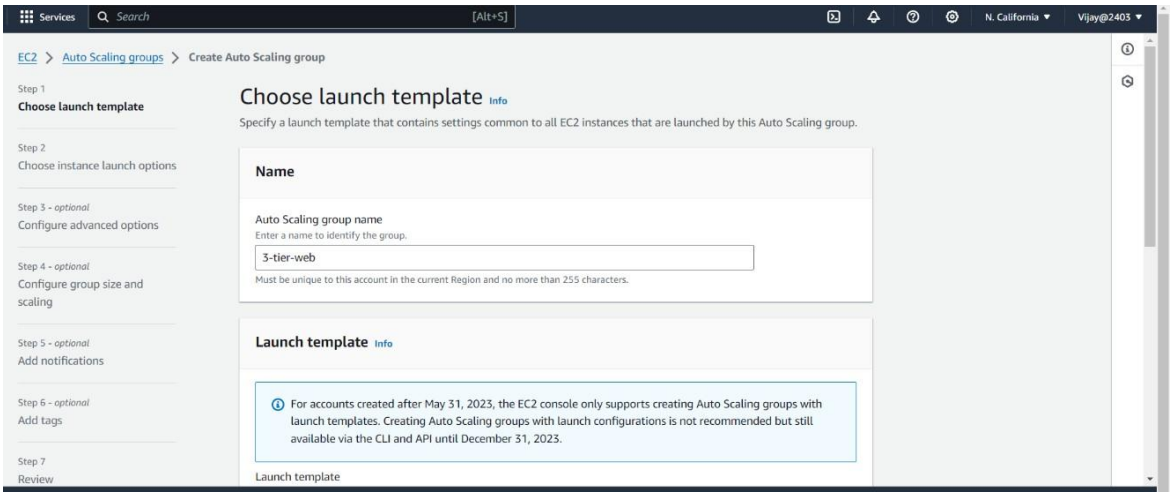


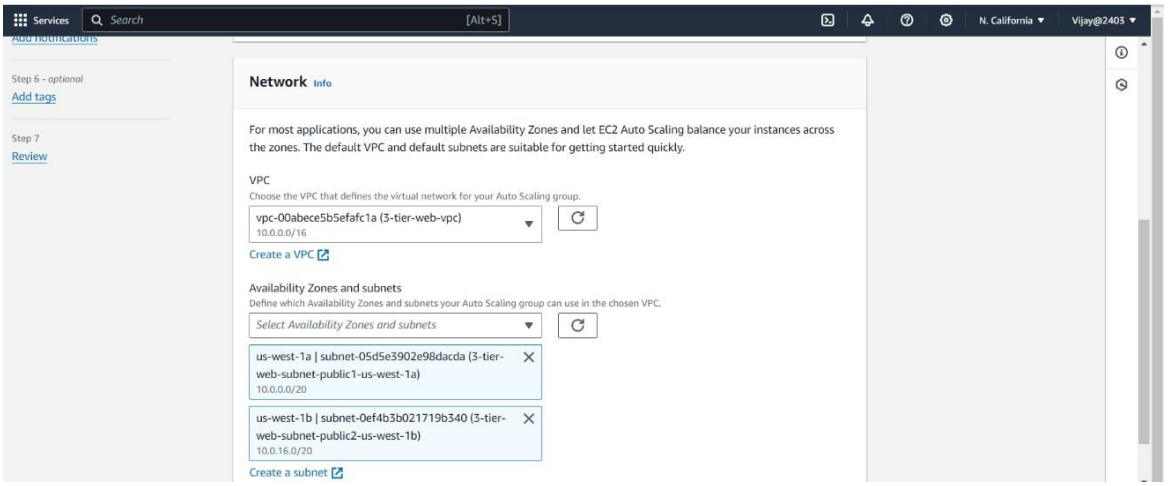




- Scroll to the bottom and paste the following script into the User Data field:

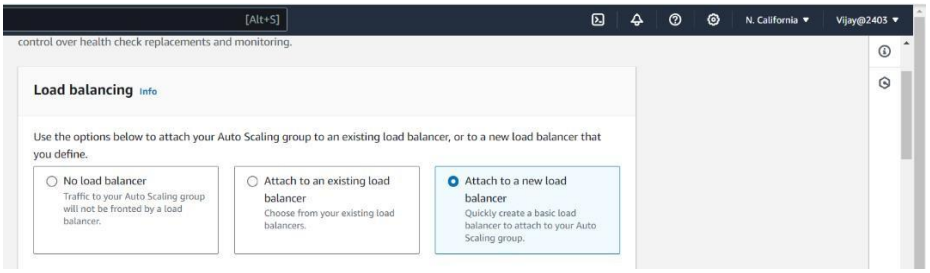
```bash
#!/bin/bash
apt update -y
apt install apache2 -y
systemctl start apache2
echo “<center><h1>vijay</h1></center>” > /var/www/html/index.html
```

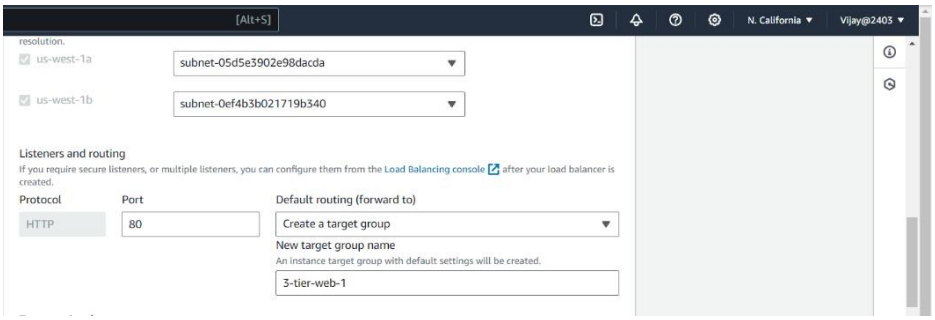
- Click the Create launch template button.
- Click the View Launch template button.
- Navigate to EC2 > Auto Scaling Groups.
- Click the Create Auto Scaling group button.
- Provide the Auto Scaling group name.




- Choose Attach a new load balancer.
- Select the Application Load Balancer type.
- Ensure that the Internet-facing Load balancer scheme is selected.
- Choose Create a target group for Default routing.
- A new target group with default settings will be created.



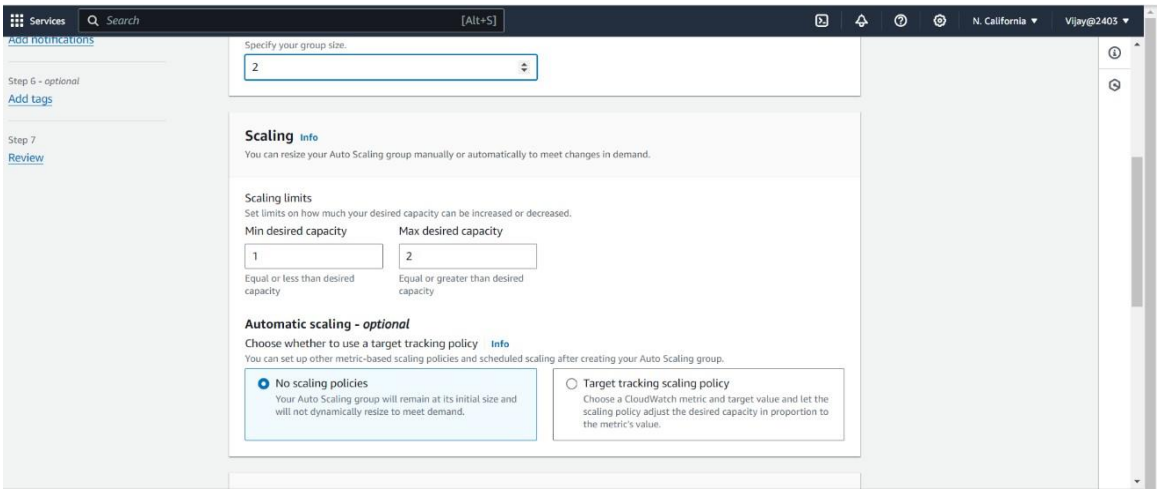



- Confirm that Health check is already enabled.

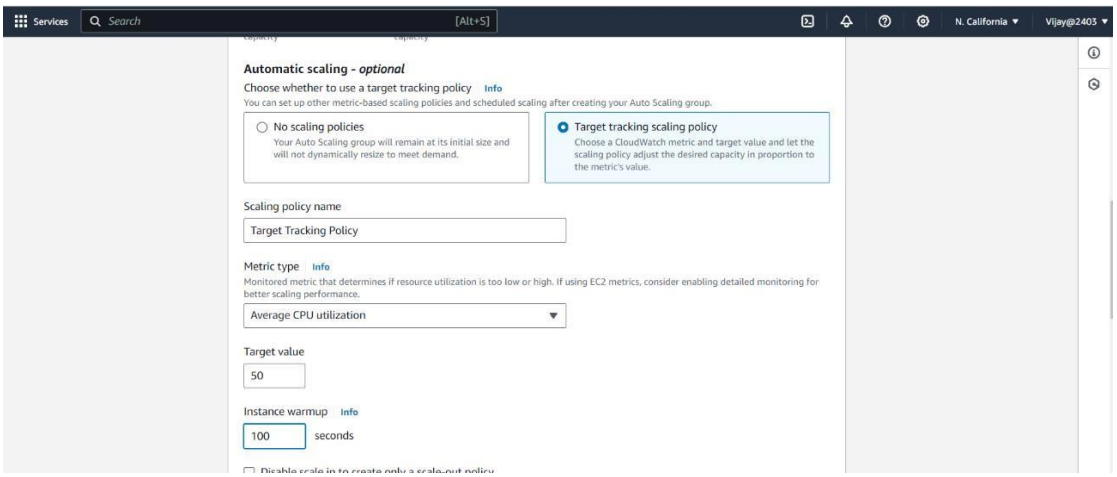
- Ensure that Enable group metrics collections with CloudWatch is already selected.

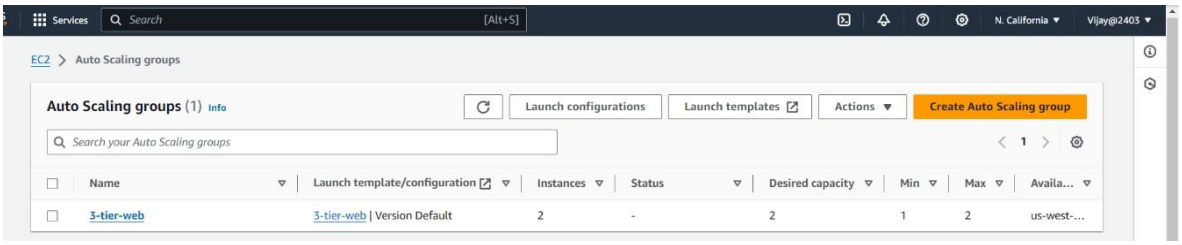
- Click the Next button.

- Specify the desired size of the Auto Scaling group.abled, and Enable group metrics collections with CloudWatch is already selected. Click Next button. Specify the desired size of the Auto Scaling group.

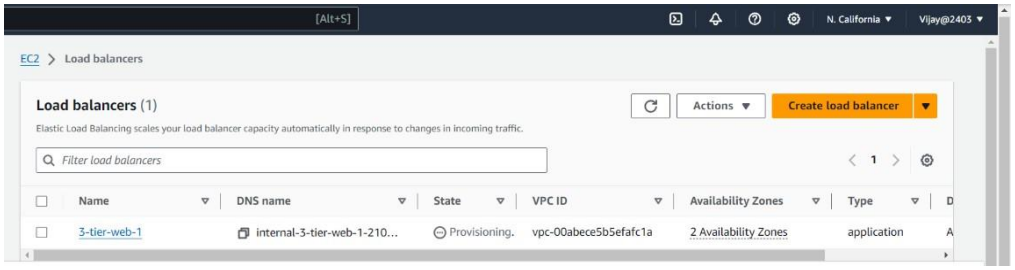



Choose Target tracking scaling policy. Click Next button.
Click Next button again. On Review page, click Create Auto Scaling group.






Go to EC2 > Load Balancers and confirm if it has been created successfully.




Go to EC2 and confirm if two EC2 instances are running. Yes, two EC2 instances are initializing. Go to a web Brower and enter the public IPv4 address. Yay! Success!

**hii instance 1**

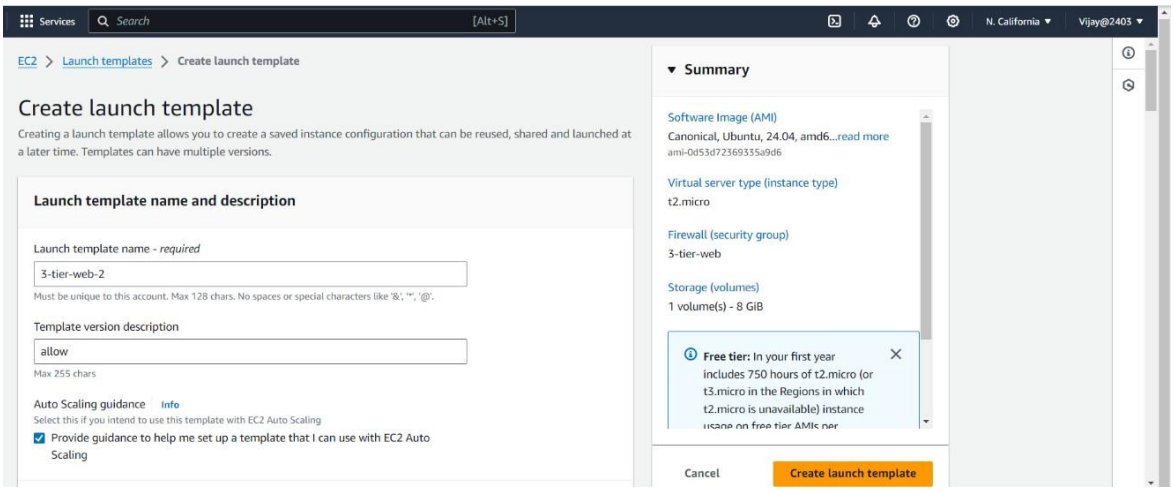
Instance-1

**hii instance 2**

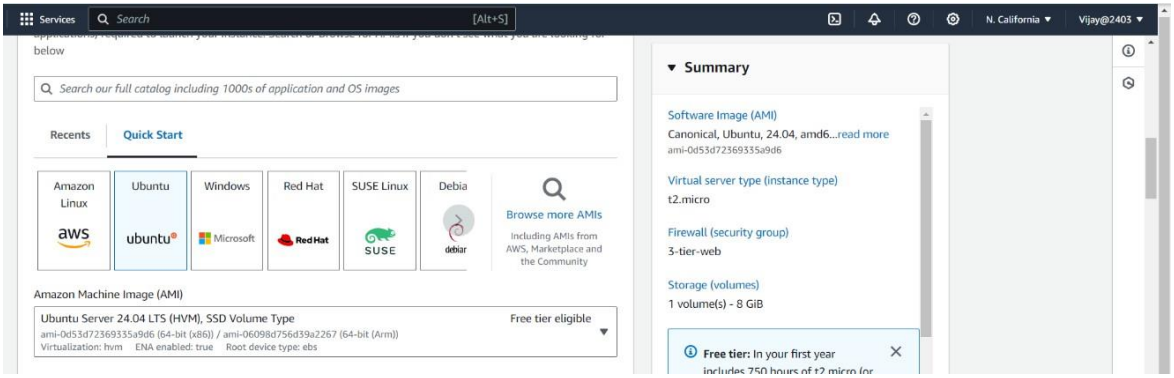
Instance-2

STEP-2: Create the App tier

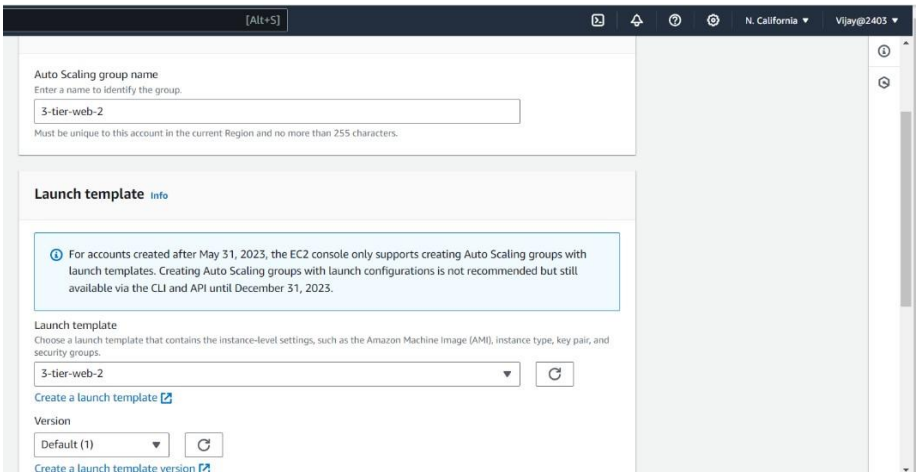
Go to EC2 > Launch templates> Click Create launch template button.
Give the Launch template name and check for Auto Scaling guidance.




Select Ubuntu server 24.04 LTS and t2.micro Instance type.

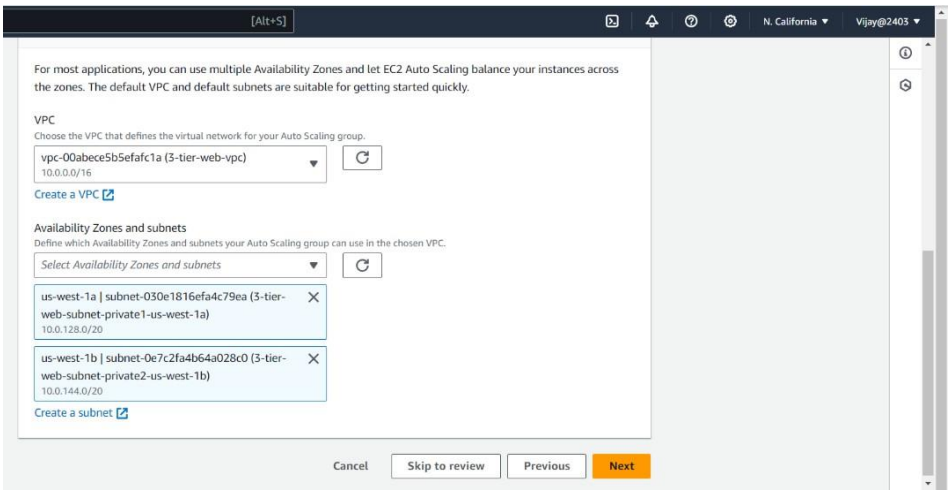


Select the Key pair you already have.



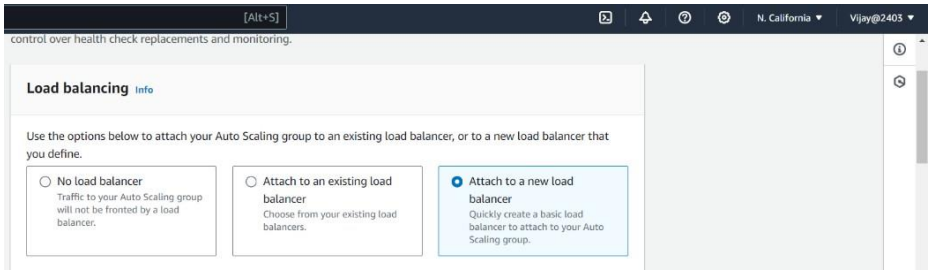

- Under Network settings, select Create security group.
- Provide a name for the security group and choose the VPC created for the tier 3 project.
- Click Add security group rule to allow traffic for ICMP to ping the app EC2 via SSH.
- Set the Web tier security group as the Source for the SSH group rule.
- Click the Create launch template button.

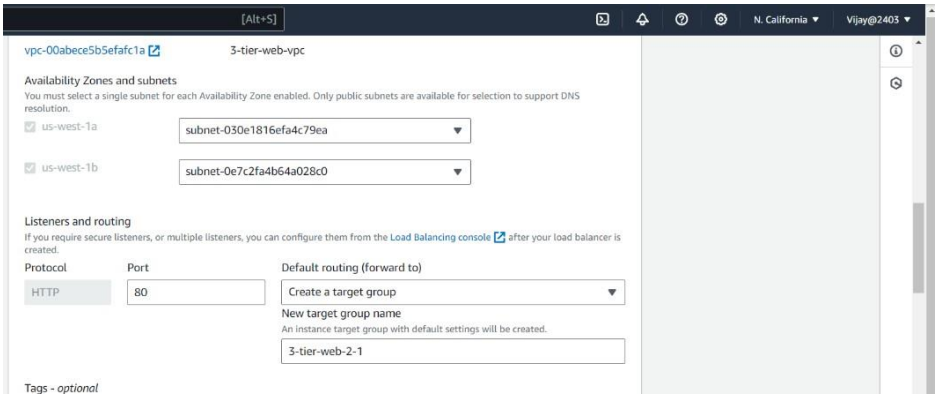
Go to EC2 > Auto Scaling groups > Click Create Auto Scaling group button. Give it a name and select the launch template for app tier and click Next button. Select VPC we created and select private subnets created for app tier. Click Next button.




Choose Attach to a new load balancer. Then, select Application Load Balancer. For app tier, we need to select Internal Load

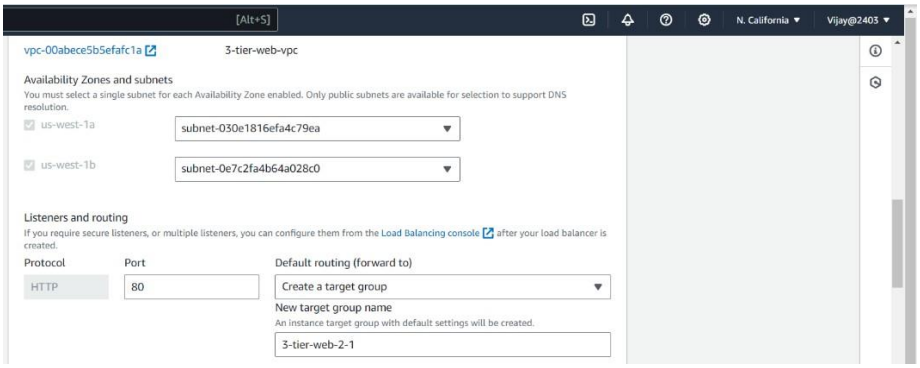
balancer scheme.



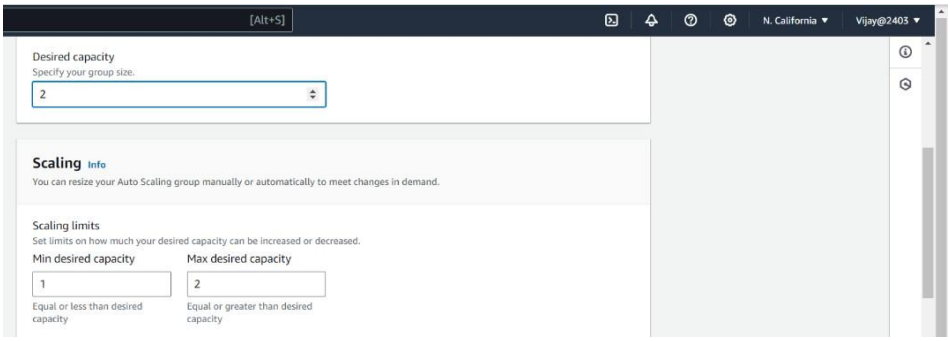




Select Create a target group. It will be automatically created for you. Keep the default settings for Health checks and

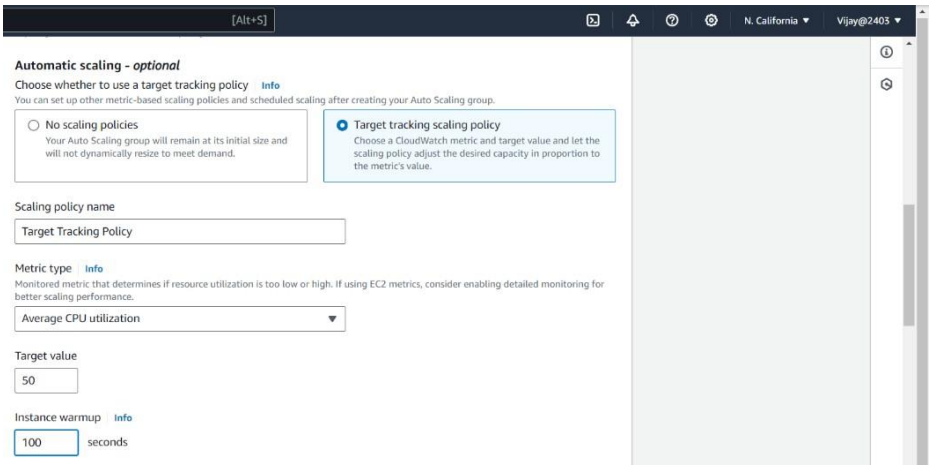
click Next button.



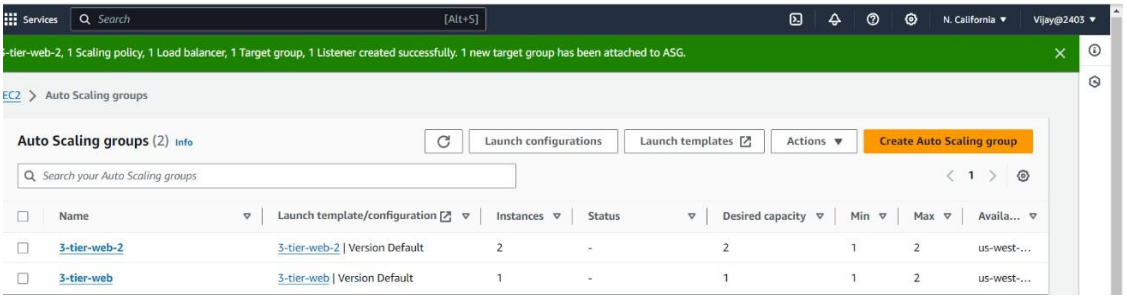
Enter your desired capacity.




Enter your Scaling policies. Then, click Next button.




Review and then click Create Auto Scaling group button.



STEP-3: Create the Database Tier

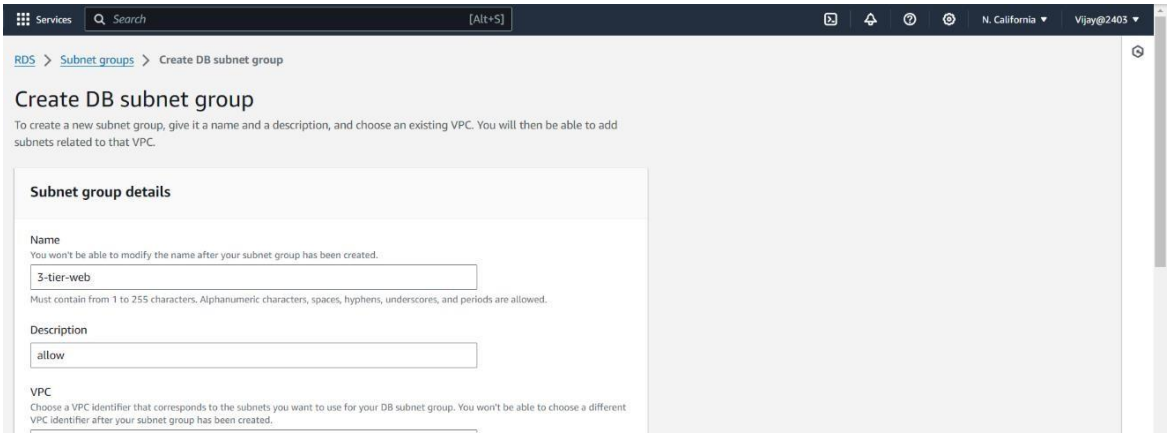
Navigate to EC2 > Auto Scaling groups.

- Click the Create Auto Scaling group button.

- Provide a name for the Auto Scaling group and select the launch template for the app tier.

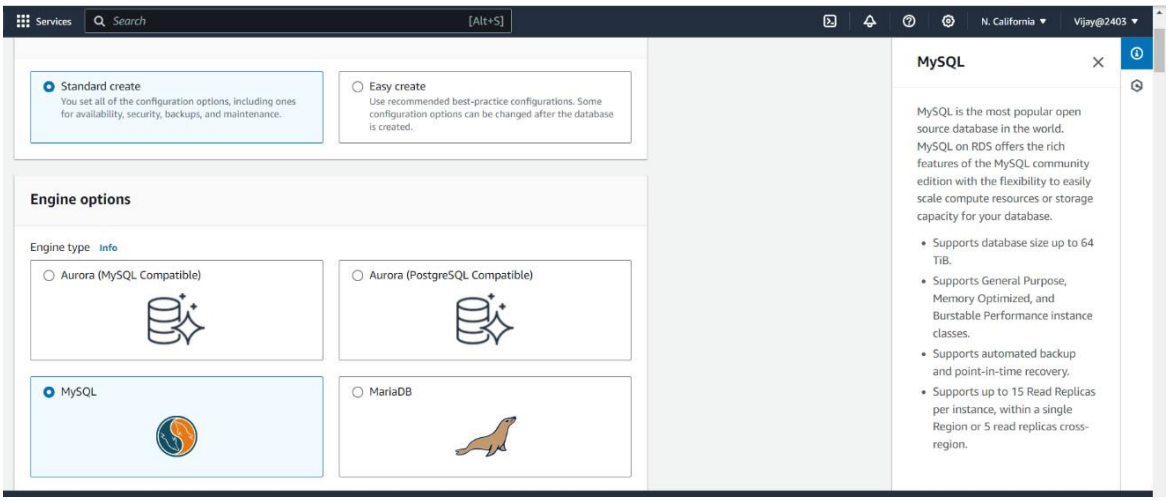
- Click the Next button.

- Select the VPC created for the project and choose the private subnets created for the app tier.

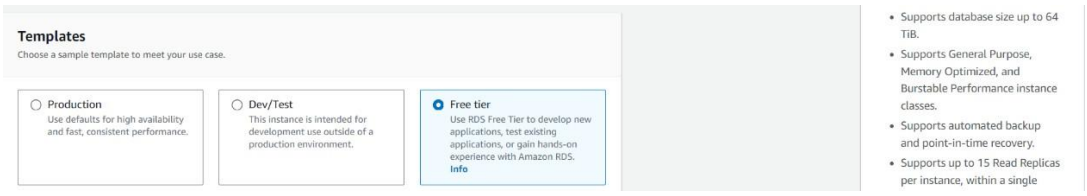


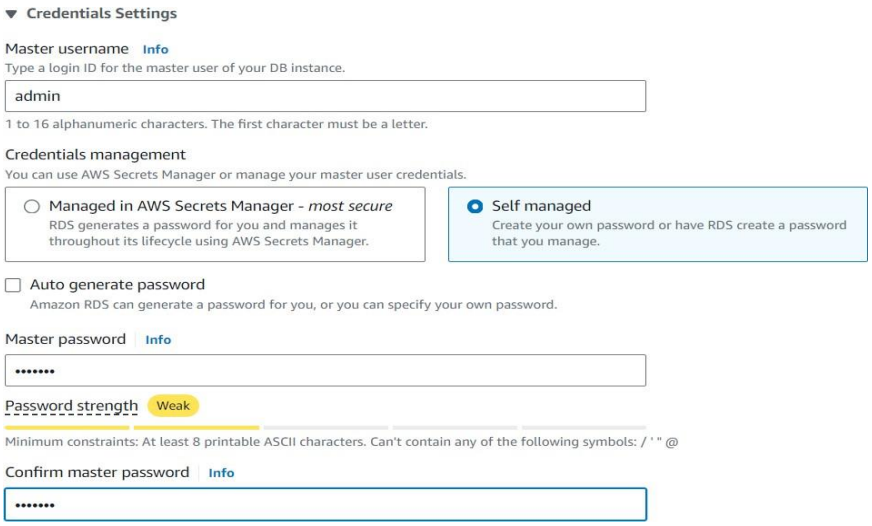

- Click the Next button.

Navigate to RDS > Click Create database button. Choose Standard create method. Select MySQL Engine type.

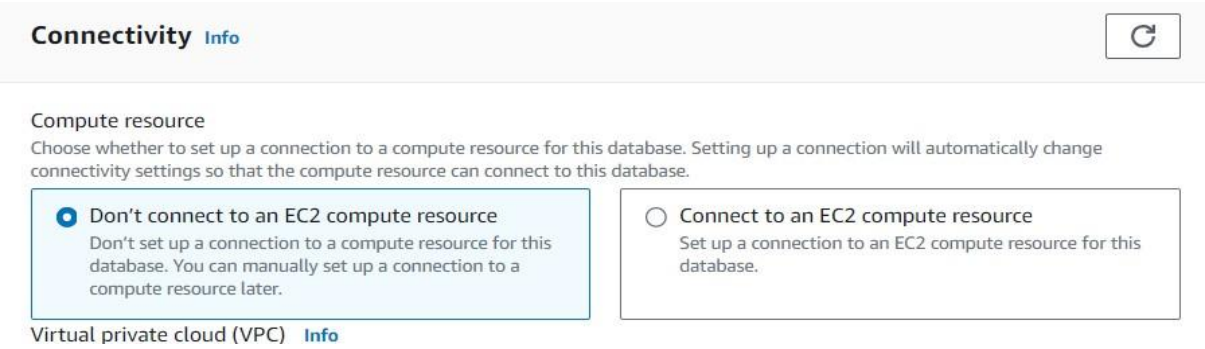



Choose Free tier Template. Give your DB instance name and set up the master credential.



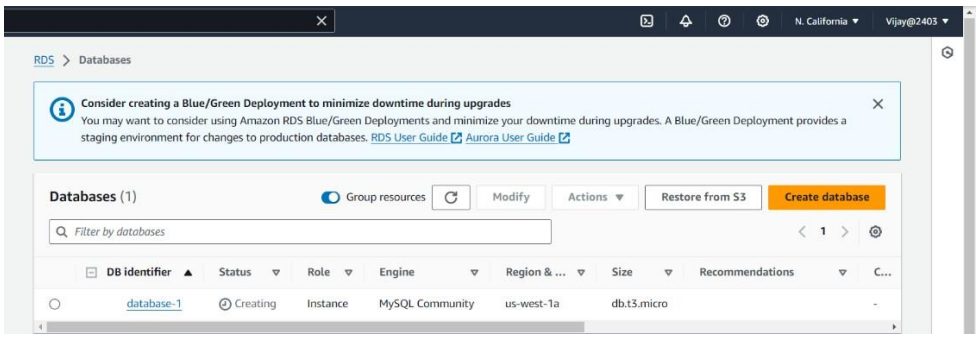



- Keep the default instance configuration and select the db.t2.micro instance type.
- Choose Don’t connect to an EC2 compute resource.
- Add the VPC created for the project.
- Select the DB subnet group you created.

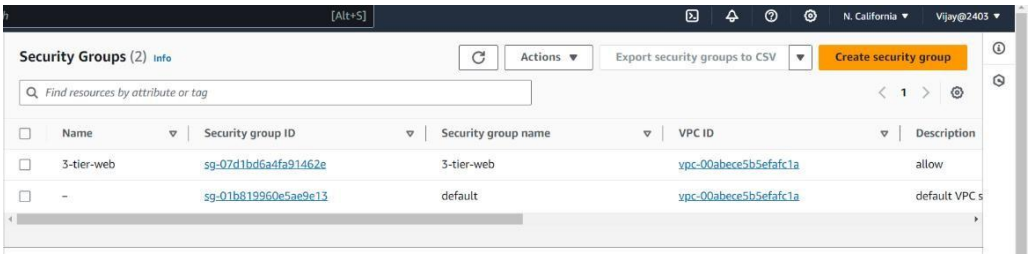



- Set Public access to No to ensure that only EC2 instances can connect to your RDS database.

Keep the default setting for the rest of sections and then click Create database button.

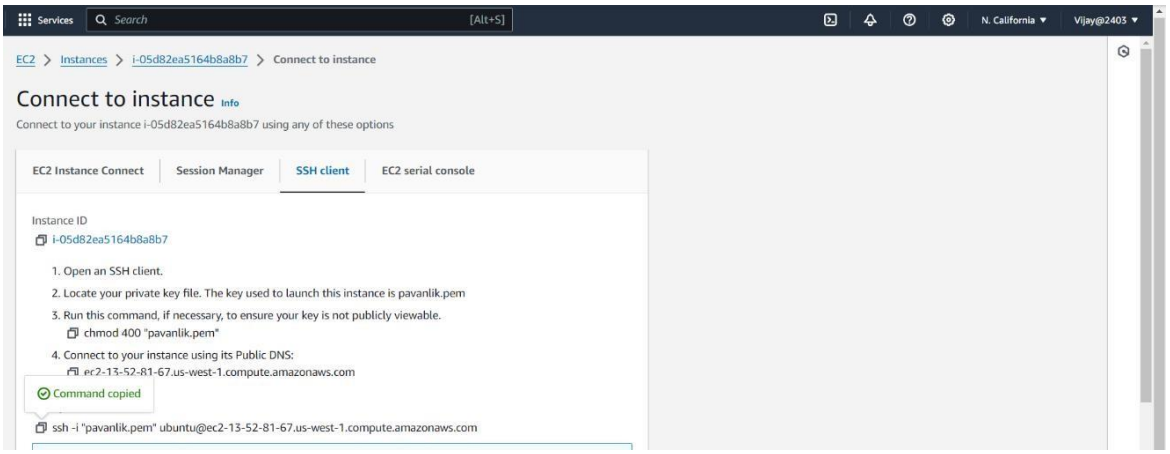


- Navigate to EC2 > Security Groups.
- Type the keyword "data" to find the matched security group.
- Select the relevant security group.
- Go to the Inbound rules tab.

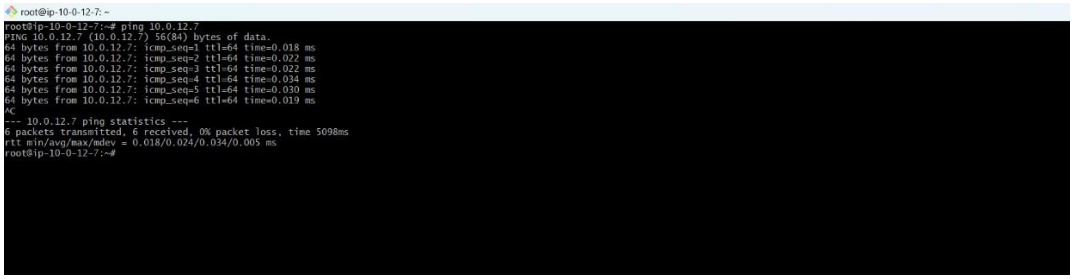



**- Click the Edit inbound rules button.**

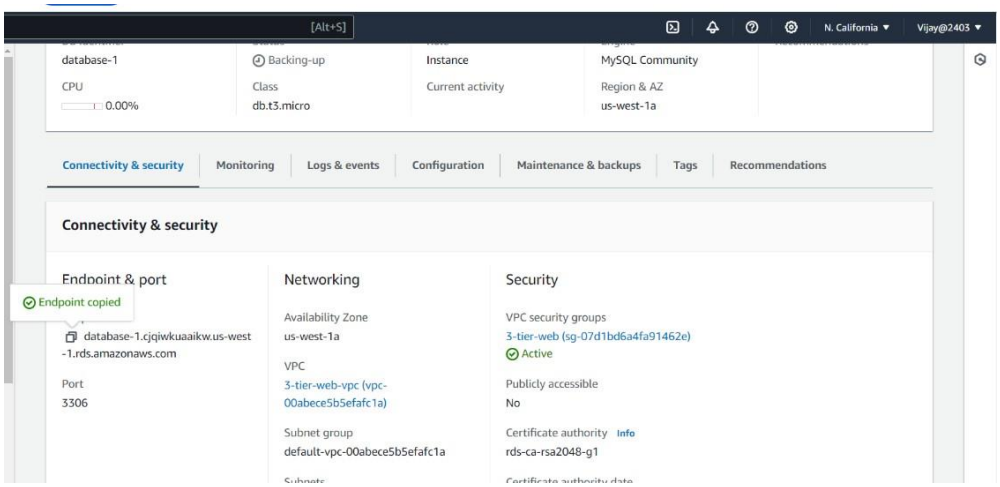
Change the Inbound Source to App Tier Security Group so that the App tier can communicate to Database Tier. Click Save rules button.
Now click Connect button to copy the ssh address.

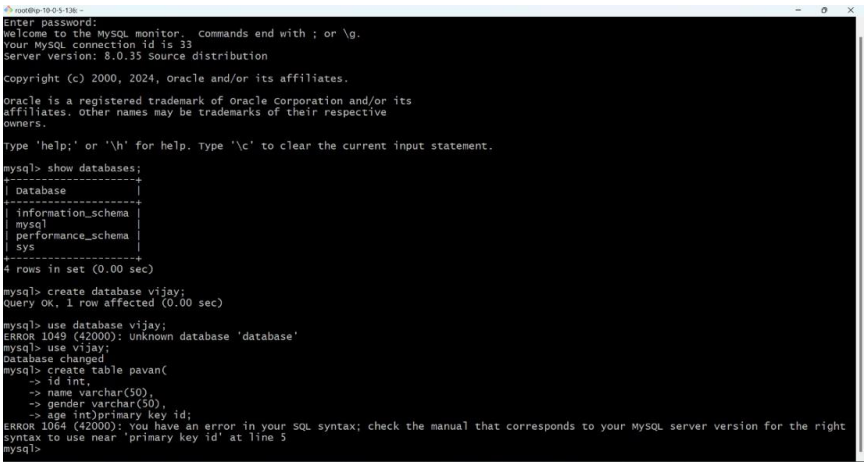
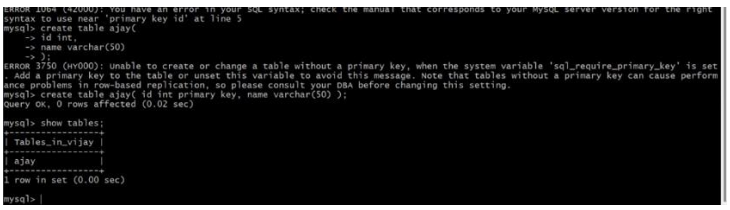



SSH into the EC2 instance. Ping one of the Private IPv4 from EC2 in the Private Subnet. Yes, I can ping the app EC2 instance private IP.



- Run the following commands to test connectivity to the Database Tier:
  ```bash
  sudo apt update
  sudo apt install mysql-server
  sudo systemctl start mysql.service
  ```
- Navigate to RDS > Databases.
- Select the Tier 3 database.
- Click the Connectivity & security tab.
- Copy the Endpoint address.




Run command below and then enter your admin password. Yay, success! Now, we are in the RDS database instance.