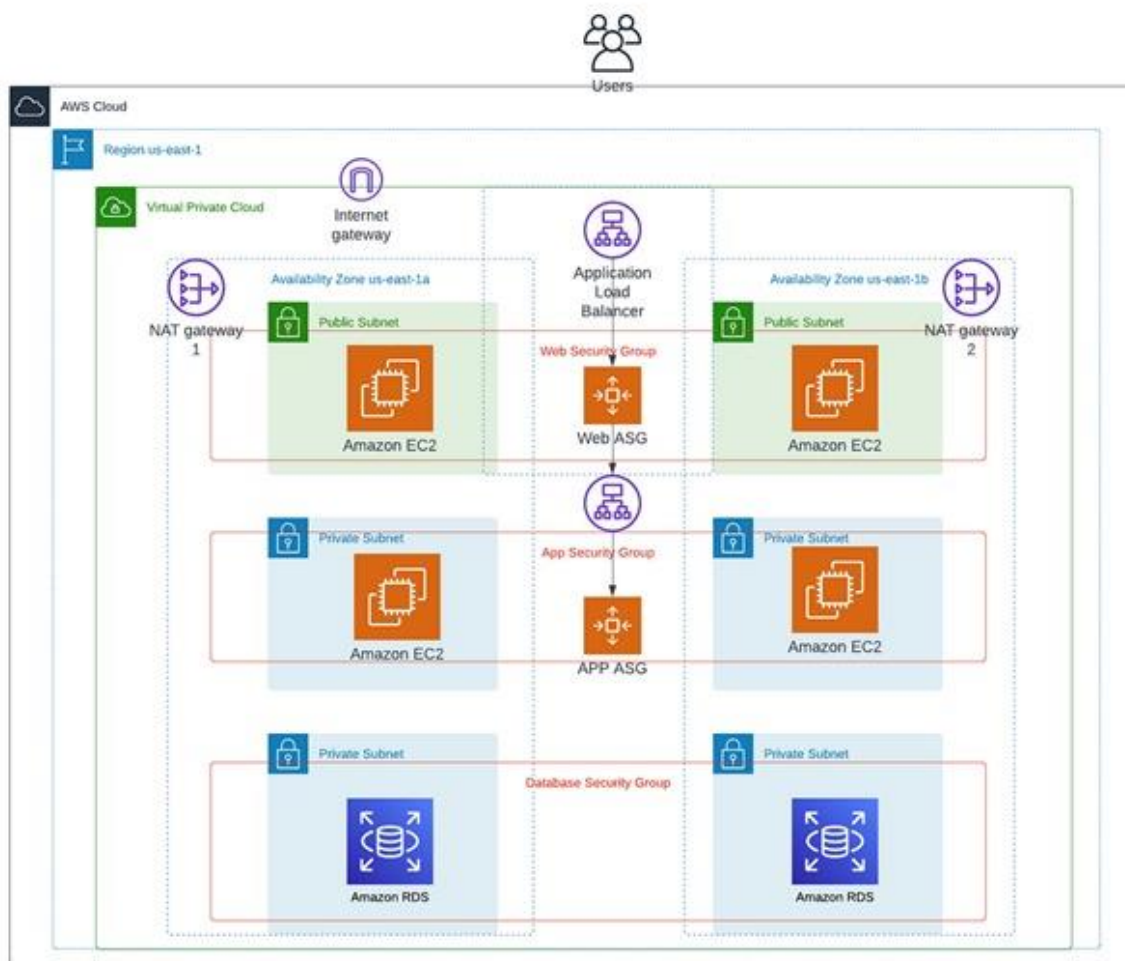
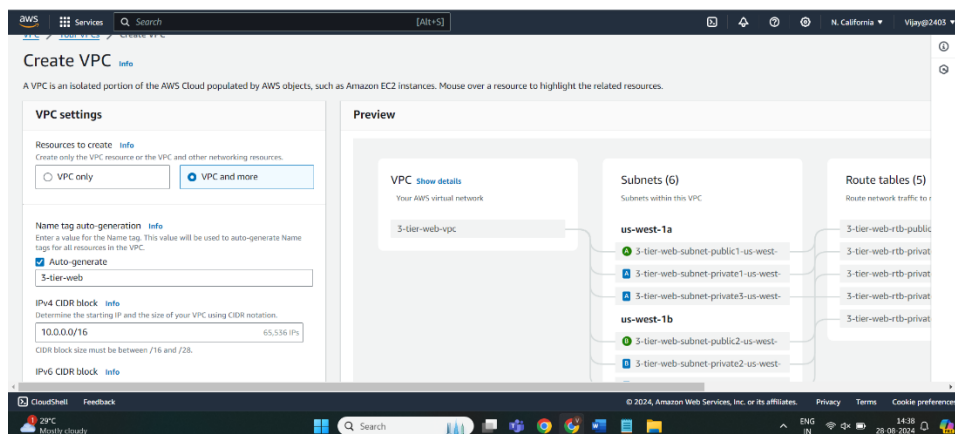


# PROJECT-1

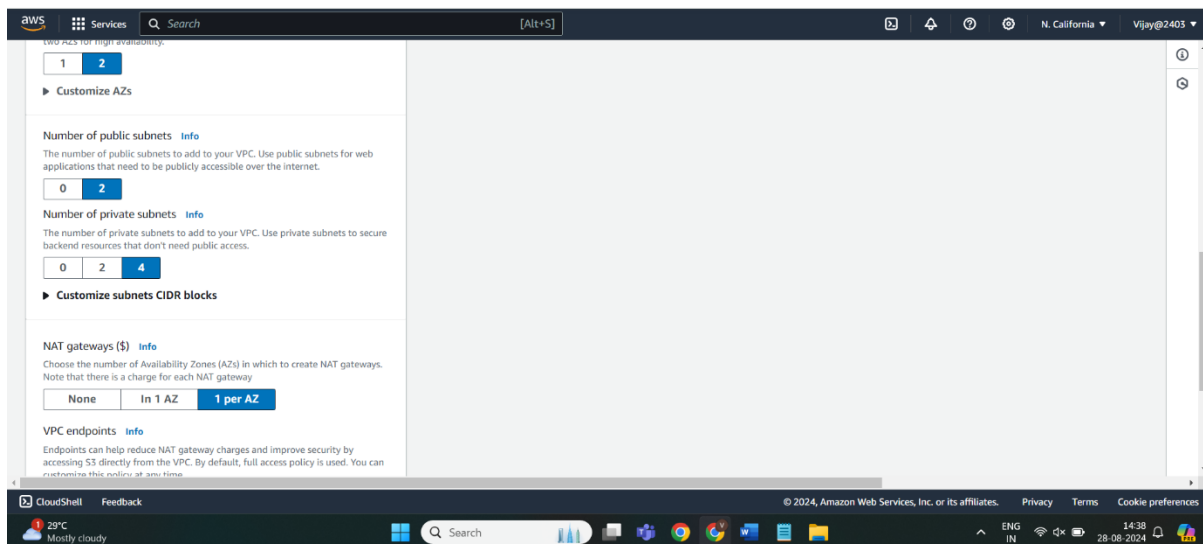
## 3-TIER ARCHITECTURE



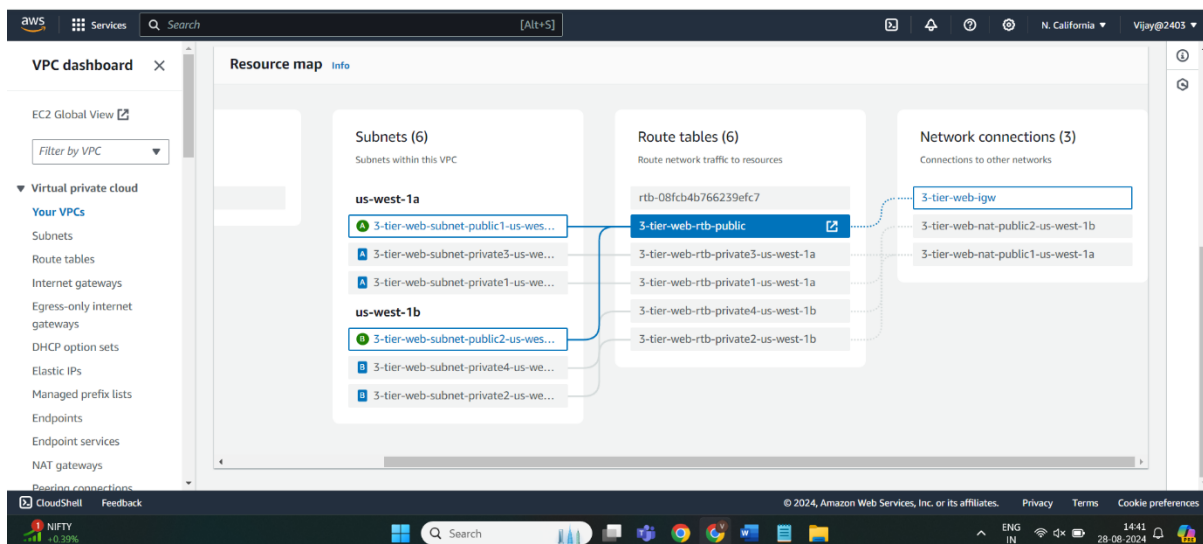
Go to VPC > Create VPC > Select VPC and more option. Then, keep the IPv4 CIDR block as 10.0.0.0/16 and click No for IPv6 CIDR block. Keep Tenancy as Default.



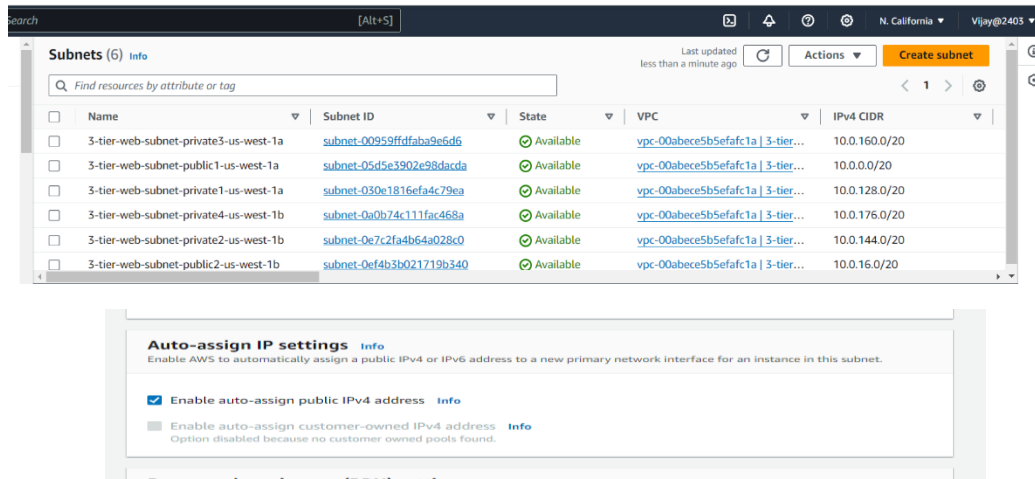
Select '2' Availability Zone and select the First availability zone as us-west-1a and the Second availability zone as us-west-1b. Enter the public and private subnet CIDR information as shown below. Choose 1 per AZ for NAT gateway and None for VPC endpoints. Enable DNS hostnames and DNS resolution. Click Create VPC button.



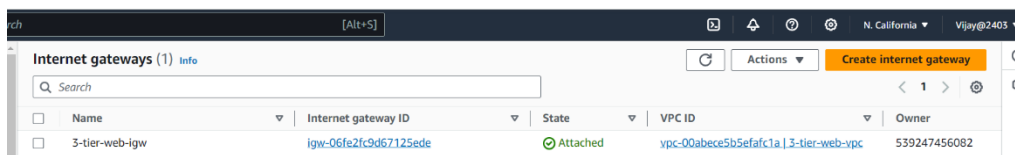
Once completed, click View VPC button. Now, we have a VPC, 2 public subnets, 4 private subnets, and 2 NAT gateways.



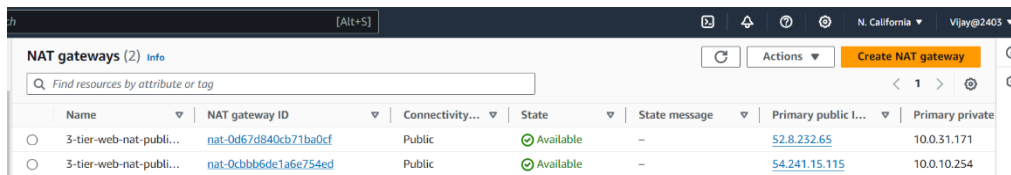
Go to VPC > Subnets > Select one of two public subnets and then click Actions menu button. Click Edit subnet settings. Enable auto-assign public IPV4 address. Click Save button. Repeat step 6 for the other public subnet.



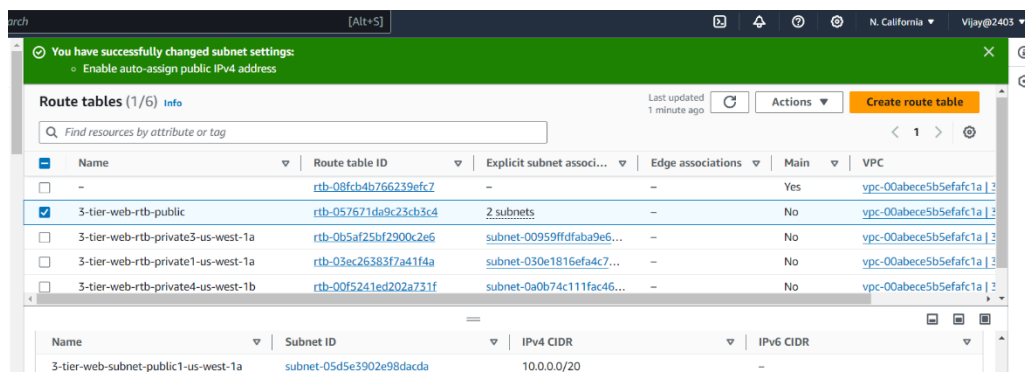
Navigate to Internet Gateway and confirm that Internet Gateway is attached to the VPC we just created.



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



Navigate to public Route tables and confirm that both public subnets are explicitly associated. Confirmed that each private route table is explicitly associated with the private subnet.



## STEP-1: CREATE WEB TIER

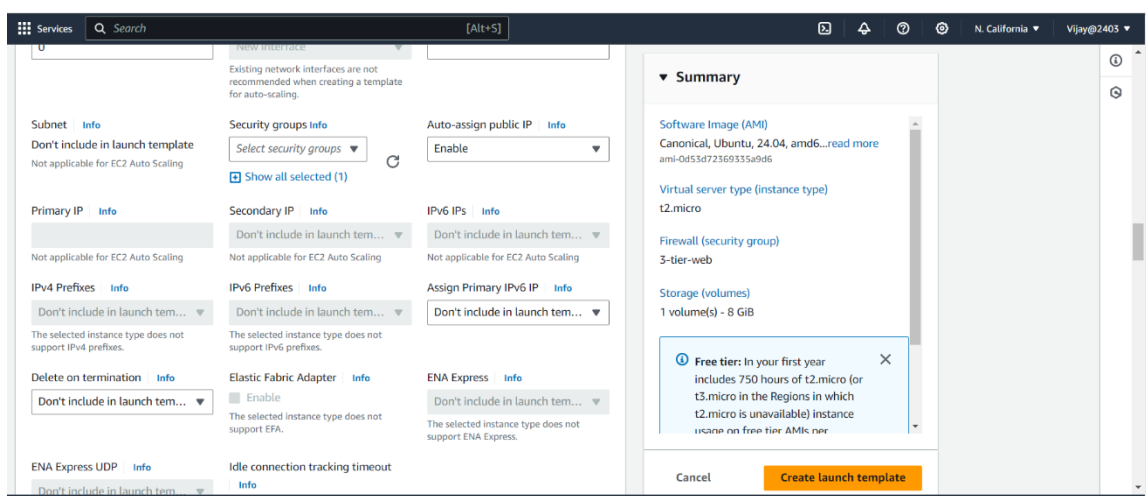
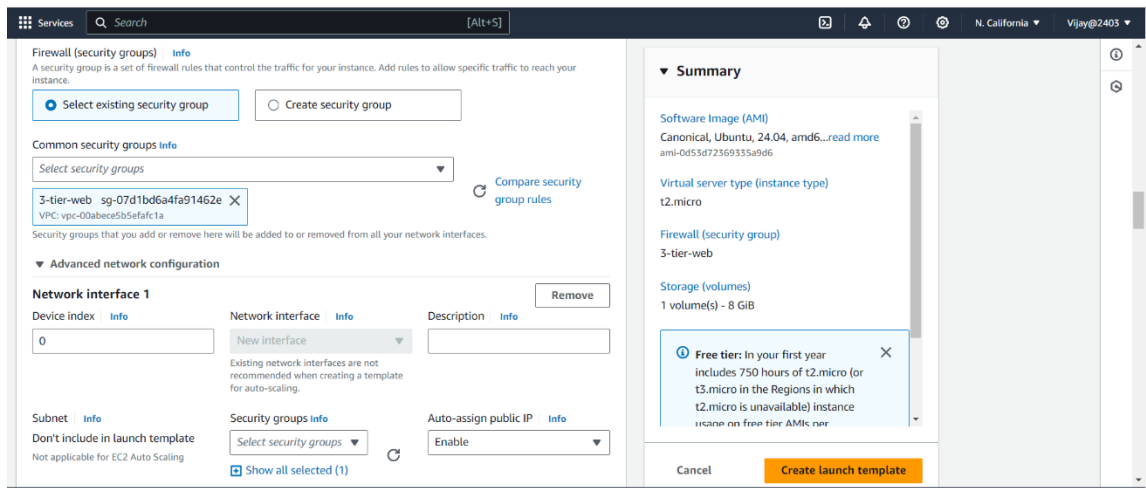
Go to EC2 > Launch Templates > Click Create launch template button. Give the launch template name and description. Check the box for Auto Scaling guidance.

The screenshot shows the 'Create launch template' page in the AWS Management Console. The page has a header with 'EC2 > Launch templates > Create launch template'. The main section is titled 'Create launch template' with a sub-header 'Launch template name and description'. It contains two input fields: 'Launch template name - required' with the value '3-tier-web' and 'Template version description' with the value 'allow'. Below these fields is a checkbox for 'Auto Scaling guidance' which is checked. To the right, a 'Summary' panel shows the configuration: 'Software Image (AMI)' as 'Canonical, Ubuntu, 24.04, amd64...read more', 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as '3-tier-web', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A blue notification box states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per user'. At the bottom right are 'Cancel' and 'Create launch template' buttons.

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

The screenshot shows the 'Application and OS images (Amazon Machine Image)' page in the AWS Management Console. It features a search bar and a 'Quick Start' section with logos for Amazon Linux, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. The 'Ubuntu' logo is highlighted. Below the logos, the 'Amazon Machine Image (AMI)' section shows 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' as the selected AMI, with a 'Free tier eligible' tag. The 'Summary' panel on the right shows the configuration: 'Software Image (AMI)' as 'Canonical, Ubuntu, 24.04, amd64...read more', 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as '3-tier-web', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A blue notification box states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per user'. At the bottom right are 'Cancel' and 'Create launch template' buttons.

Select security group and give it a name and choose the VPC we created in Part 1. Click Add Security Group Rules button and add rules to allow ports for ICMP, HTTP, and SSH as shown below. Under Advanced network configuration, enable Auto-assign public IP.



Scroll to the bottom and add below script to the User Data field.

```
#!/bin/bash
```

```
apt update -y
```

```
apt install apache2 -y
```

```
systemctl start apache2
```

```
echo "<center><h1>vijay</h1></center>" >
```

```
var/www/html/index.html
```

Click Create launch template button. Click View Launch template button.

Go to EC2 >Auto Scaling Groups > Click Create Auto Scaling group and give the Auto Scaling group name.

Services Search [Alt+S] N. California Vijay@2403

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template

Step 2  
Choose instance launch options

Step 3 - optional  
Configure advanced options

Step 4 - optional  
Configure group size and scaling

Step 5 - optional  
Add notifications

Step 6 - optional  
Add tags

Step 7  
Review

## Choose launch template [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

**Name**

Auto Scaling group name  
Enter a name to identify the group.

3-tier-web

Must be unique to this account in the current Region and no more than 255 characters.

**Launch template [Info](#)**

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template

Services Search [Alt+S] N. California Vijay@2403

Step 6 - optional  
[Add tags](#)

Step 7  
[Review](#)

## Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**  
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-00abece5b5efafc1a (3-tier-web-vpc)  
10.0.0.0/16

[Create a VPC](#)

**Availability Zones and subnets**  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-west-1a | subnet-05d5e3902e98dacda (3-tier-web-subnet-public1-us-west-1a)  
10.0.0.0/20

us-west-1b | subnet-0ef4b3b021719b340 (3-tier-web-subnet-public2-us-west-1b)  
10.0.16.0/20

[Create a subnet](#)

**Now it's time to configure load balancer to distribute the traffic. Choose Attach a new load balancer. Choose Application Load Balancer type and make sure you select Internet-facing Load balancer scheme. Select Create a target group for Default routing. A new target group with default settings will be created.**

[Alt+S] N. California Vijay@2403

control over health check replacements and monitoring.

## Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☐ Attach to an existing load balancer  
Choose from your existing load balancers.

☒ Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

resolution.

us-west-1a subnet-05d5e3902e98dacda

us-west-1b subnet-0ef4b3b021719b340

**Listeners and routing**

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group

New target group name

An instance target group with default settings will be created.

3-tier-web-1

**Health check is already enabled, and Enable group metrics collections with CloudWatch is already selected. Click Next button.**

**Specify the desired size of the Auto Scaling group.**

Specify your group size.

2

**Scaling** [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

**Scaling limits**

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
1	2

Equal or less than desired capacity

Equal or greater than desired capacity

**Automatic scaling - optional**

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☒ **No scaling policies**

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☐ **Target tracking scaling policy**

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

**Choose Target tracking scaling policy. Click Next button.**

**Click Next button again. On Review page, click Create Auto Scaling group.**

**Automatic scaling - optional**

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☐ **No scaling policies**

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☒ **Target tracking scaling policy**

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name

Target Tracking Policy

**Metric type** [Info](#)

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

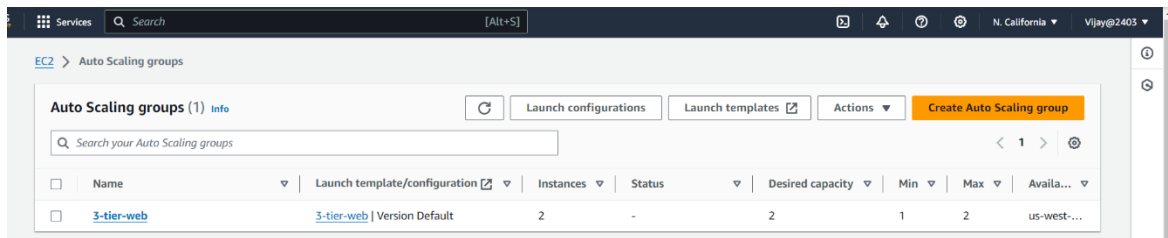
Target value

50

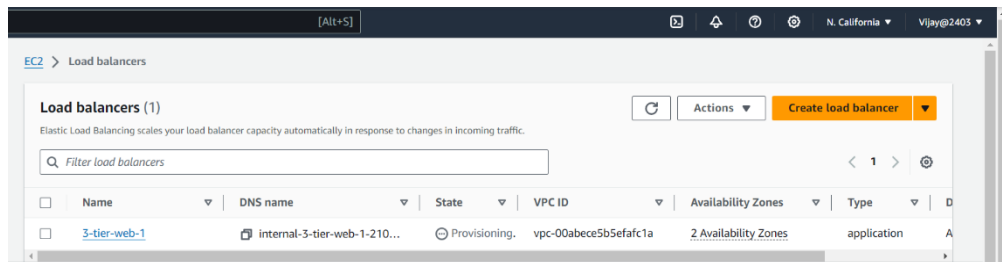
Instance warmup [Info](#)

100 seconds

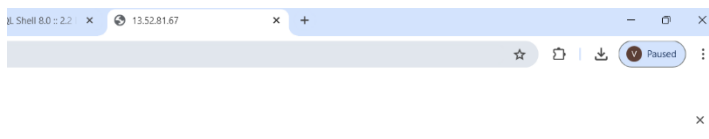
☐ Disable scale-in to create only a scale-out policy



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

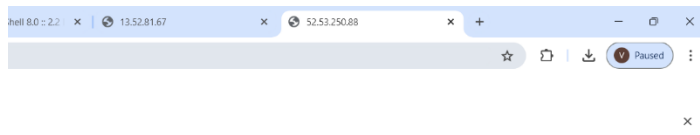


**Go to EC2 and confirm if two EC2 instances are running. Yes, two EC2 instances are initializing. Go to a web Browser and enter the public IPv4 address. Yay! Success!**



**VIJAY**

**Instance-1**



**VIJAY**

**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.

The screenshot shows the 'Create launch template' page in the AWS Management Console. The page is divided into two main sections: 'Launch template name and description' on the left and 'Summary' on the right. In the left section, the 'Launch template name' field is filled with '3-tier-web-2' and the 'Template version description' field is filled with 'allow'. Below these fields, there is a checkbox for 'Auto Scaling guidance' which is checked. The right section, titled 'Summary', lists the configuration details: 'Software Image (AMI)' as 'Canonical, Ubuntu, 24.04, amd64...', 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as '3-tier-web', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A blue callout box in the summary section states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.' At the bottom right, there are 'Cancel' and 'Create launch template' buttons.

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

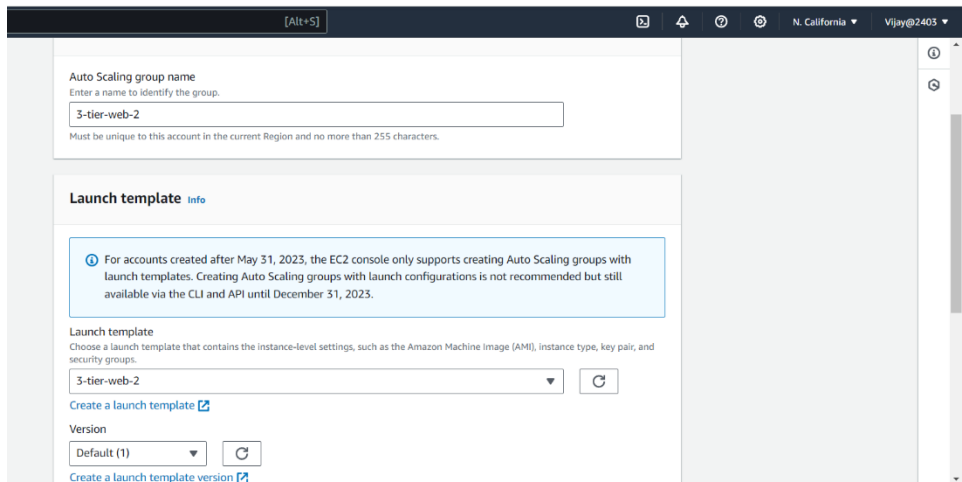
The screenshot shows the 'Launch templates' page in the AWS Management Console. The page has a search bar at the top and a 'Quick Start' section below it. The 'Quick Start' section displays a grid of operating system logos: Amazon Linux, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. Below the grid, there is a section for 'Amazon Machine Image (AMI)' which shows 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' as the selected option. To the right of this section, there is a 'Free tier eligible' label. The right side of the page shows the 'Summary' section, which lists the configuration details: 'Software Image (AMI)' as 'Canonical, Ubuntu, 24.04, amd64...', 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as '3-tier-web', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A blue callout box in the summary section states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.'

Select the Key pair you already have.

The screenshot shows the 'Launch template' page in the AWS Management Console. The page is divided into two main sections: 'Auto Scaling group name' at the top and 'Launch template' below it. In the 'Auto Scaling group name' section, the field is filled with '3-tier-web-2'. The 'Launch template' section has a blue callout box stating: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' Below this, there is a dropdown menu for 'Launch template' which is set to '3-tier-web-2'. There are also links for 'Create a launch template' and 'Create a launch template version'. At the bottom, there is a 'Version' section with a dropdown menu set to 'Default (1)' and a 'Create a launch template version' link.

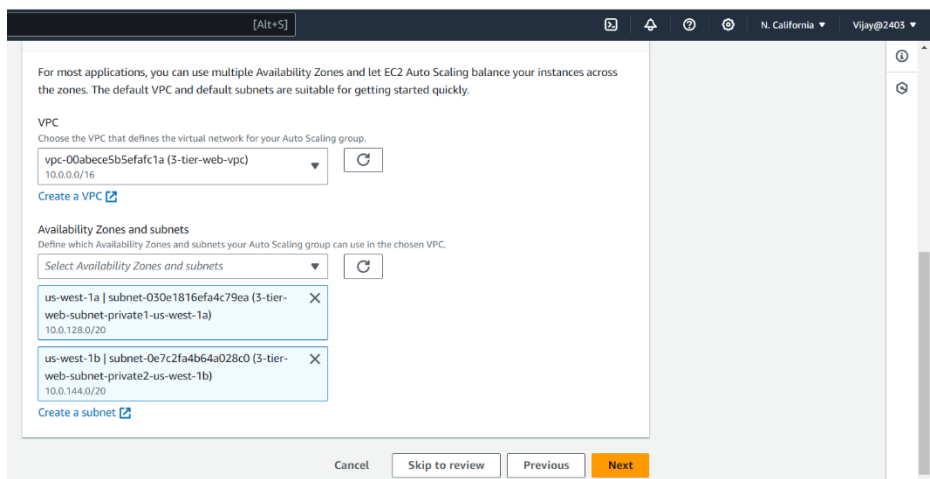
**Under Network settings, select Create security group and give it a name and choose the VPC we created for tier 3 project. Click Add security group rule to allow ports for ICMP to ping the app EC2 via SSH. Choose Web tier security group as Source for ssh group rule.**

**Click Create launch template button. And then, click View Launch template.**



The screenshot shows the 'Create Auto Scaling group' page in the AWS Management Console, specifically the 'Network' section. The 'Auto Scaling group name' is '3-tier-web-2'. A message states: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' The 'Launch template' dropdown is set to '3-tier-web-2'. The 'Version' dropdown is set to 'Default (1)'. There are links to 'Create a launch template' and 'Create a launch template version'.

**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.**



The screenshot shows the 'Create Auto Scaling group' page in the AWS Management Console, specifically the 'VPC and subnets' section. The 'VPC' dropdown is set to 'vpc-00abece5b5efafc1a (3-tier-web-vpc)'. The 'Availability Zones and subnets' section shows two selected subnets: 'us-west-1a | subnet-030e1816ef4c79ea (3-tier-web-subnet-private1-us-west-1a)' and 'us-west-1b | subnet-0e7c2fa4b64a028c0 (3-tier-web-subnet-private2-us-west-1b)'. At the bottom, there are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

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

[Alt+S] N. California Vijay@2403

control over health check replacements and monitoring.

### Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☐ Attach to an existing load balancer  
Choose from your existing load balancers.

☒ Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

[Alt+S] N. California Vijay@2403

vpc-00abece5b5efafc1a 3-tier-web-vpc

### Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☒ us-west-1a subnet-030e1816efa4c79ea

☒ us-west-1b subnet-0e7c2fa4b64a028c0

### Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group

New target group name  
An instance target group with default settings will be created.

3-tier-web-2-1

Tags - optional

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

[Alt+S] N. California Vijay@2403

vpc-00abece5b5efafc1a 3-tier-web-vpc

### Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☒ us-west-1a subnet-030e1816efa4c79ea

☒ us-west-1b subnet-0e7c2fa4b64a028c0

### Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group

New target group name  
An instance target group with default settings will be created.

3-tier-web-2-1

Enter your desired capacity.

The screenshot shows the 'Desired capacity' step in the AWS Auto Scaling console. The 'Desired capacity' field is set to 2. Below it, the 'Scaling limits' section shows 'Min desired capacity' set to 1 and 'Max desired capacity' set to 2. The interface includes a top navigation bar with the user's name 'Vijay@2403' and a sidebar with navigation links.

Desired capacity  
Specify your group size.

2

Scaling [Info](#)  
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits  
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity  
1  
Equal or less than desired capacity

Max desired capacity  
2  
Equal or greater than desired capacity

Enter your Scaling policies. Then, click Next button.

The screenshot shows the 'Automatic scaling - optional' step in the AWS Auto Scaling console. The 'Target tracking scaling policy' is selected. The 'Scaling policy name' is 'Target Tracking Policy'. The 'Metric type' is 'Average CPU utilization' and the 'Target value' is 50. The 'Instance warmup' is set to 100 seconds. The interface includes a top navigation bar with the user's name 'Vijay@2403' and a sidebar with navigation links.

Automatic scaling - optional  
Choose whether to use a target tracking policy [Info](#)  
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

☐ No scaling policies  
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

☒ Target tracking scaling policy  
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name  
Target Tracking Policy

Metric type [Info](#)  
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value  
50

Instance warmup [Info](#)  
100 seconds

Review and then click Create Auto Scaling group button.

The screenshot shows the 'Review' step in the AWS Auto Scaling console. A green banner at the top indicates that the scaling policy, load balancer, target group, and listener have been created successfully. Below the banner, the 'Auto Scaling groups (2)' section displays a table with two groups: '3-tier-web-2' and '3-tier-web'. The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability zones. The 'Create Auto Scaling group' button is visible in the top right corner.

Services [Search](#) [Alt+S] N. California Vijay@2403

3-tier-web-2, 1 Scaling policy, 1 Load balancer, 1 Target group, 1 Listener created successfully. 1 new target group has been attached to ASG.

EC2 > Auto Scaling groups

Auto Scaling groups (2) [Info](#) [Refresh](#) [Launch configurations](#) [Launch templates](#) [Actions](#) [Create Auto Scaling group](#)

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availa...
<input type="checkbox"/>	3-tier-web-2	3-tier-web-2   Version Default	2	-	2	1	2	us-west...
<input type="checkbox"/>	3-tier-web	3-tier-web   Version Default	1	-	1	1	2	us-west...

## STEP-3: Create the Database Tier

Navigate to Amazon RDS> Subnet groups > Click Create DB subnet group button. Give a name and choose the VPC we created. Add the availability zones we used for Web Tier and App Tier and select the two private subnets created for Database Tier. Click Create button.

The screenshot shows the 'Create DB subnet group' page in the Amazon RDS console. The breadcrumb navigation is 'RDS > Subnet groups > Create DB subnet group'. The page title is 'Create DB subnet group'. Below the title, there is a brief instruction: 'To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.'

The 'Subnet group details' section contains three fields:

- Name:** A text input field containing '3-tier-web'. Below the field, a note states: 'You won't be able to modify the name after your subnet group has been created. Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.'
- Description:** A text input field containing 'allow'.
- VPC:** A dropdown menu with the text 'Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.'

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

The screenshot shows the 'Create database' page in the Amazon RDS console. The breadcrumb navigation is 'RDS > Create database'. The page title is 'Create database'. Below the title, there are two tabs: 'Standard create' (selected) and 'Easy create'. The 'Standard create' tab has a description: 'You set all of the configuration options, including ones for availability, security, backups, and maintenance.' The 'Easy create' tab has a description: 'Use recommended best-practice configurations. Some configuration options can be changed after the database is created.'

The 'Engine options' section contains four options:

- Aurora (MySQL Compatible):** A radio button option with a database icon.
- Aurora (PostgreSQL Compatible):** A radio button option with a database icon.
- MySQL:** A radio button option, currently selected, with a MySQL logo icon.
- MariaDB:** A radio button option with a MariaDB logo icon.

On the right side of the page, there is a sidebar for the selected 'MySQL' engine type. It contains the following text: 'MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.' Below this text, there is a list of features:

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

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

The screenshot shows the 'Create database' page in the Amazon RDS console, specifically the 'Templates' section. The breadcrumb navigation is 'RDS > Create database'. The page title is 'Create database'. Below the title, there is a brief instruction: 'Choose a sample template to meet your use case.'

The 'Templates' section contains three options:

- Production:** A radio button option with a description: 'Use defaults for high availability and fast, consistent performance.'
- Dev/Test:** A radio button option with a description: 'This instance is intended for development use outside of a production environment.'
- Free tier:** A radio button option, currently selected, with a description: 'Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info'

On the right side of the page, there is a sidebar for the selected 'Free tier' template. It contains the following text: 'MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.' Below this text, there is a list of features:

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

#### ▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

#### Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

☐ Managed in AWS Secrets Manager - *most secure*  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ Self managed  
Create your own password or have RDS create a password that you manage.

☐ Auto generate password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

\*\*\*\*\*

Password strength [Weak](#)

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password [Info](#)

\*\*\*\*\*

**Keep the default Instance configuration and select db.t2.micro instance. Choose Don't connect to an EC2 compute resource and add VPC we created. Choose the DB subnet group we created. Select No for Public access so that only EC2 can connect to your RDS database.**

#### Connectivity [Info](#)



#### Compute resource

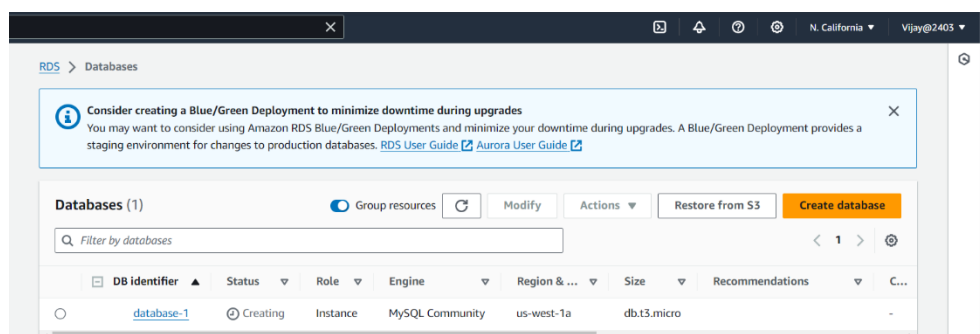
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☒ Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

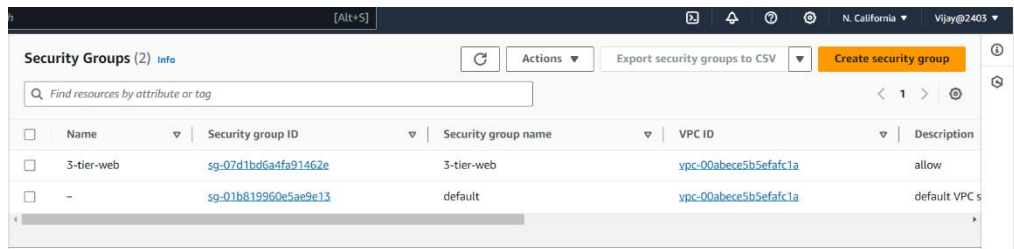
☐ Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) [Info](#)

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

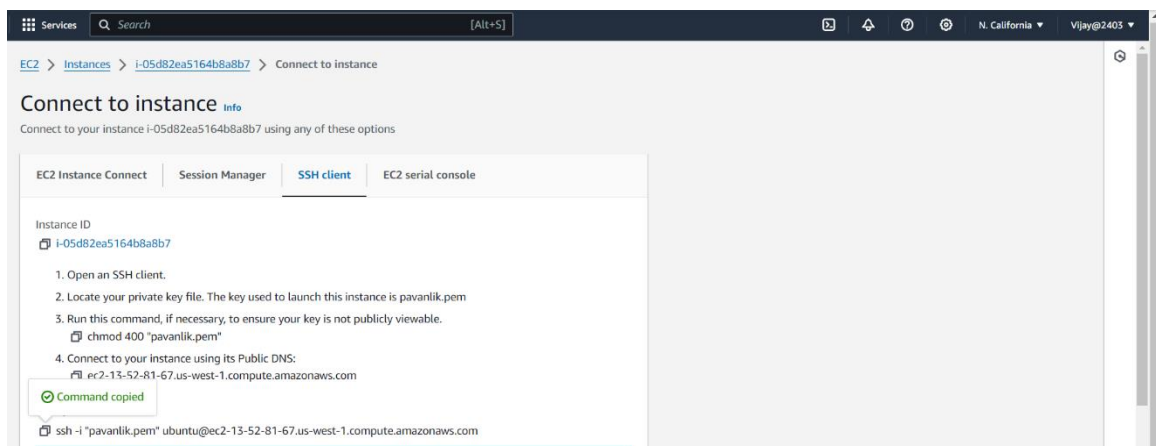


Go to EC2>Security Groups to verify the Inbound rules. Type key word data and the matched security group will show up. Select the security group. Go to Inbound rules tab. Click Edit inbound rules button.



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.



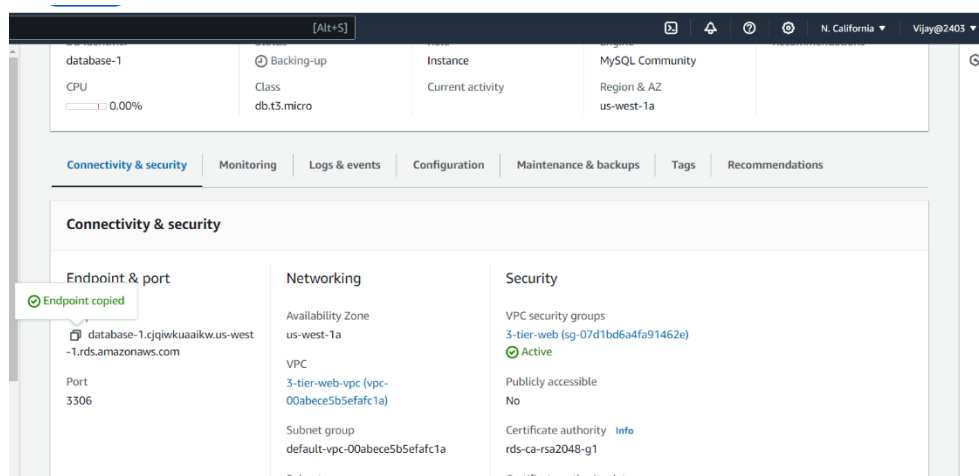
Now let's test connectivity to the Database Tier. Run

**#sudo apt update**

**#sudo apt install mysql-server**

**#sudo systemctl start mysql.service**

Go to RDS > Database > Tier 3 database > Click Connectivity & security tab and copy the Endpoint address.



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

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database vijay;
Query OK, 1 row affected (0.00 sec)

mysql> use database vijay;
ERROR 1049 (42000): Unknown database 'database'
mysql> use vijay;
Database changed
mysql> create table pavan(
  -> id int,
  -> name varchar(50),
  -> gender varchar(50),
  -> age int)primary key id;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'primary key id' at line 5
mysql>

mysql> create table ajay(
  -> id int,
  -> name varchar(50)
  -> );
ERROR 3750 (HY000): unable to create or change a table without a primary key, when the system variable 'sql_require_primary_key' is set
. Add a primary key to the table or unset this variable to avoid this message. Note that tables without a primary key can cause perform
mysql> create table ajay( id int primary key, name varchar(50) );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_vijay |
+-----+
| ajay |
+-----+
1 row in set (0.00 sec)

mysql>
```