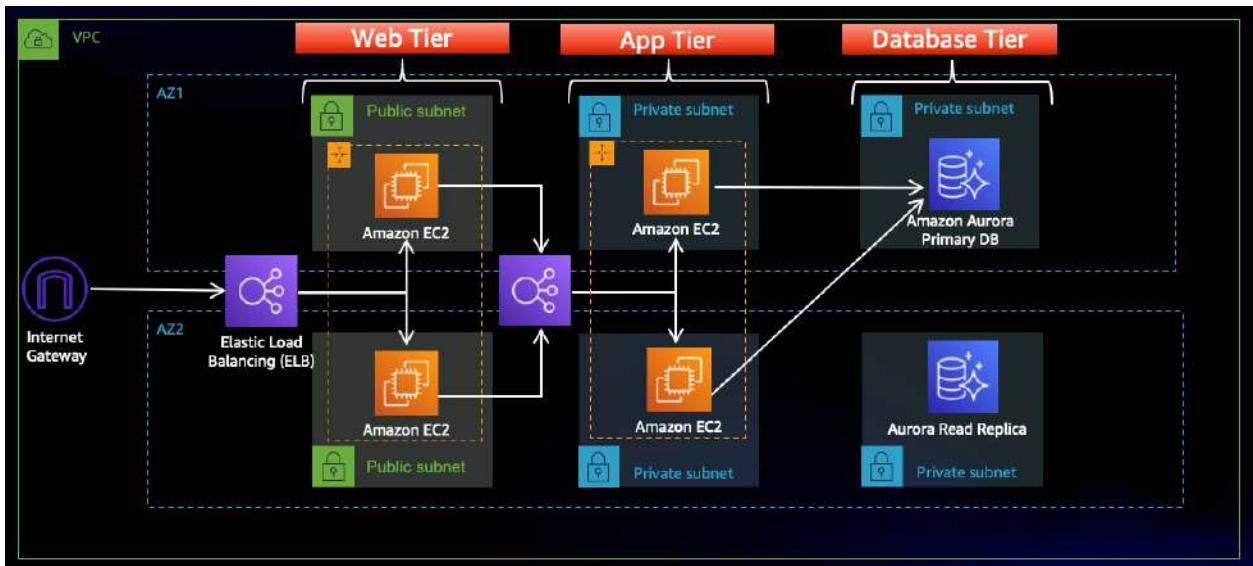


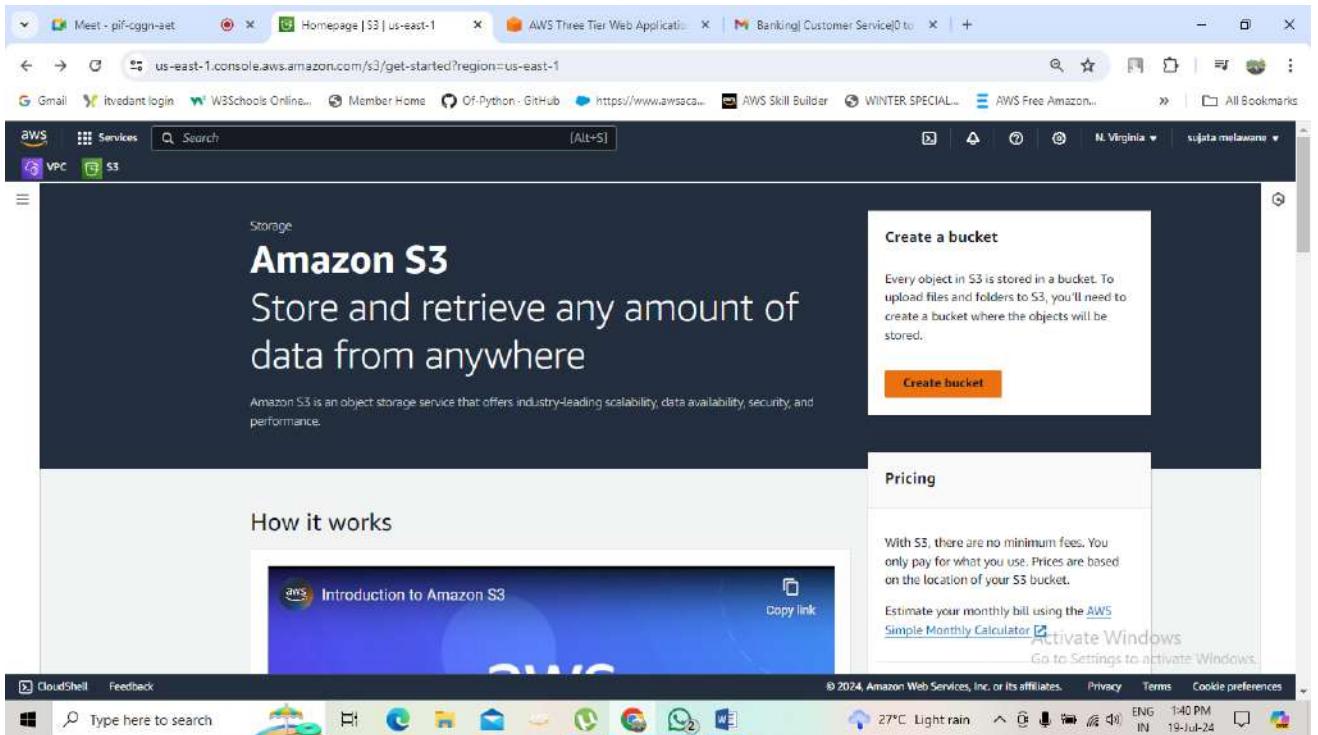
3-Tier Web Application Project

AWS Three Tier Web Architecture

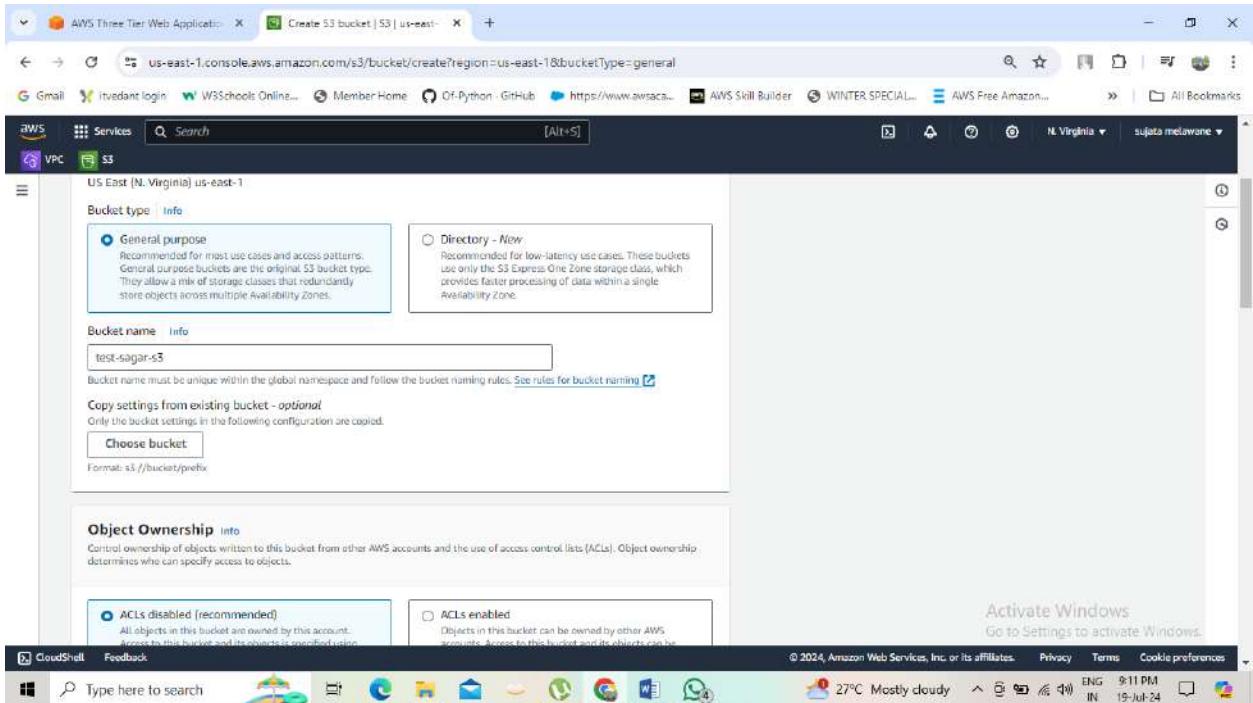


In this architecture, a public-facing Application Load Balancer forwards client traffic to our web tier EC2 instances. The web tier is running Nginx webservers that are configured to serve a React.js website and redirects our API calls to the application tier's internal facing load balancer. The internal facing load balancer then forwards that traffic to the application tier, which is written in Node.js. The application tier manipulates data in an Aurora MySQL multi-AZ database and returns it to our web tier. Load balancing, health checks and autoscaling groups are created at each layer to maintain the availability of this architecture.

1. Navigate to the S3 service in the AWS console and create a new S3 bucket.



2. Give it a unique name, and then leave all the defaults as in. Make sure to select the region that you intend to run this whole lab in. This bucket is where we will upload our code later.



The screenshot shows the AWS S3 console with a green success message: "Successfully created bucket: 'test-sagar-s3'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there's an "Account snapshot - updated every 24 hours" section and a table of "General purpose buckets".

Name	AWS Region	IAM Access Analyzer	Creation date
test-sagar-s3	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 19, 2024, 21:11:55 (UTC+05:30)

At the bottom, there's a Windows taskbar with various icons and a weather widget showing 27°C Mostly cloudy.

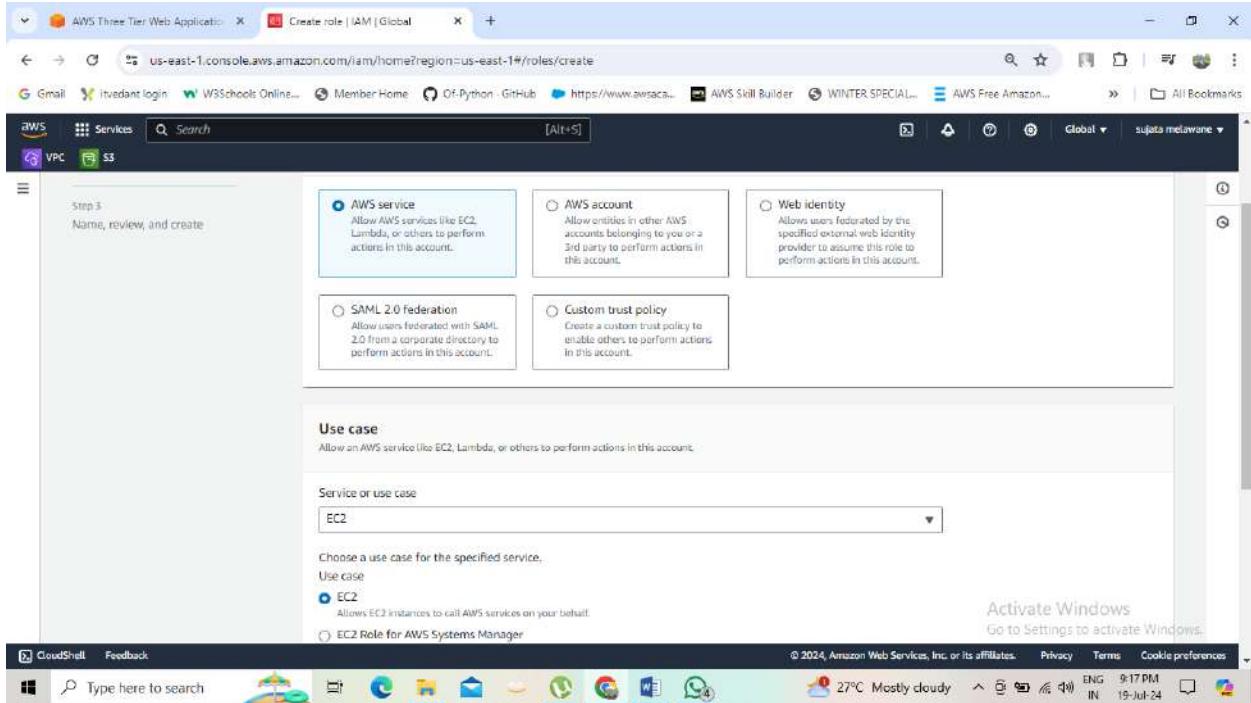
3. Navigate to the IAM dashboard in the AWS console and create an EC2 role.

The screenshot shows the AWS IAM console with the "Roles" page selected. It displays two roles: "AWSServiceRoleForSupport" and "AWSServiceRoleForTrustedAdvisor".

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-

Below the roles, there are sections for "Access AWS from your non AWS workloads", "X.509 Standard", and "Temporary credentials". At the bottom, there's a Windows taskbar with various icons and a weather widget showing 27°C Mostly cloudy.

4. Select EC2 as the trusted entity



5. When adding permissions, include the following AWS managed policies. You can search for them and select them. These policies will allow our instances to download our code from S3 and use Systems Manager Session Manager to securely connect to our instances without SSH keys through the AWS console.

- **AmazonSSMManagedInstanceCore**
- **AmazonS3ReadOnlyAccess**

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (2/943) Info

Choose one or more policies to attach to your new role.

Filter by Type

Q: AmazonSSMManagedInstanceCore X All types ▾ 1 match

Policy name ▾ Type Description

AmazonSSMManagedInstanceCore AWS managed The policy for Amazon EC2 Role to enable...

Set permissions boundary - *optional*

Cancel Previous Next

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (2/943) Info

Choose one or more policies to attach to your new role.

Filter by Type

Q: AmazonS3ReadOnlyAccess X All types ▾ 1 match

Policy name ▾ Type Description

AmazonS3ReadOnlyAccess AWS managed Provides read only access to all buckets vi...

Set permissions boundary - *optional*

Cancel Previous Next

1. Give your role a name, and then click **Create Role**.

AWS Three Tier Web Application | Create role | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#roles/create?trustedEntityType=AWS_SERVICE&selectedService=EC2&selectedUser=

Gmail Itvedant login W3Schools Online Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks

Sujieta melawane

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
Test-Rol

Description
Add a short explanation for this role.
Allows EC2 Instances to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy
AmazonSSMManagedInstanceCore	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

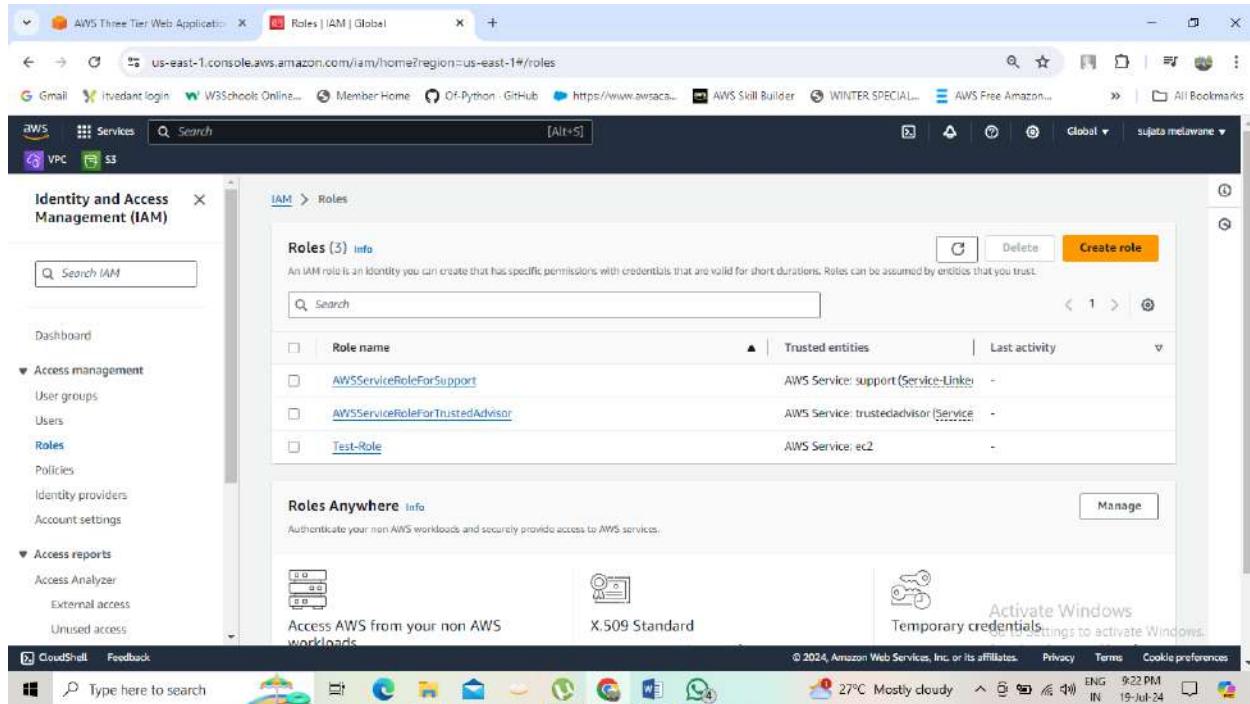
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Type here to search 27°C Mostly cloudy ENG 9:21 PM IN 19-Jul-24



Part 1: Networking and Security

In this section we will be building out the VPC networking components as well as security groups that will add a layer of protection around our EC2 instances, Aurora databases, and Elastic Load Balancers.

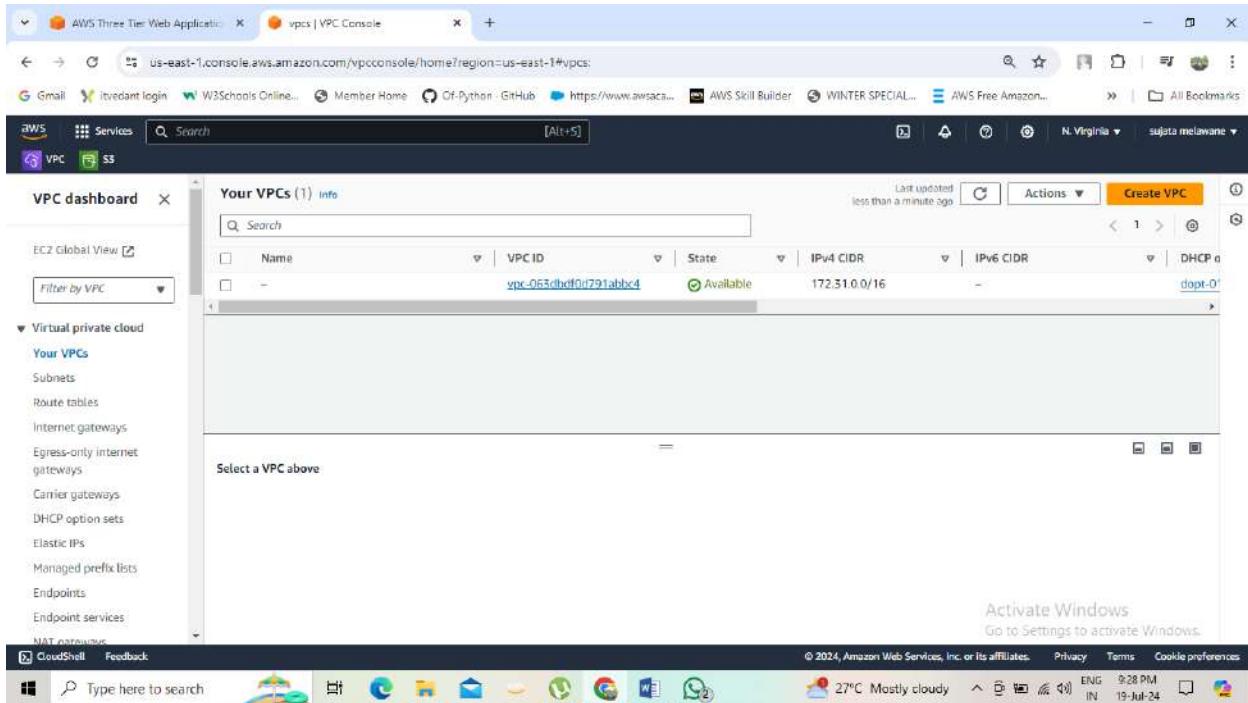
Learning Objectives:

- Create an isolated network with the following components:
 - VPC
 - Subnets
 - Route Tables
 - Internet Gateway
 - NAT gateway
 - Security Groups

VPC and Subnets

1. VPC Creation

1. Navigate to the VPC dashboard in the AWS console and navigate to **Your VPCs** on the left hand side.



The screenshot shows the AWS VPC Console interface. The top navigation bar includes tabs for 'AWS Three Tier Web Application' and 'vpcs | VPC Console'. The main content area is titled 'Your VPCs (1) info'. A table lists one VPC entry:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options
-	vpc-063d9df0f791abb4	Available	172.31.0.0/16	-	opt-0'

The left sidebar has a 'Virtual private cloud' section with various sub-options like 'Your VPCs', 'Subnets', 'Route tables', etc. The bottom status bar shows the date and time as '19-Jul-24' and '9:28 PM'.

2. Make sure **VPC only** is selected, and fill out the VPC Settings with a **Name tag** and a **CIDR range** of your choice.

NOTE: Make sure you pay attention to the region you're deploying all your resources in. You'll want to stay consistent for this workshop.

NOTE: Choose a CIDR range that will allow you to create at least 6 subnets.

AWS Three Tier Web Application CreateVpc | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1>CreateVpc:createMode=vpcOnly

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create: [Info](#)
Create only the VPC resource or the VPC and other networking resources.
 VPC only VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
test-vpc-sagar

IPv4 CIDR block: [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block
IPv4 CIDR: 10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block: [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 9:31 PM IN 19-Jul-24

Type here to search

AWS Three Tier Web Application CreateVpc | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1/CreateVpc:createMode=vpcOnly

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

10.0.0.0/16 CIDR block size must be between /16 and /28.

IPv6 CIDR block: [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy: [Info](#)
Default

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: Name Value - optional: test-vpc-sagar Remove tag Add tag You can add 49 more tags

Create VPC Cancel

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 9:32 PM IN 19-Jul-24

Type here to search

The screenshot shows the AWS VPC console interface. On the left, a sidebar titled 'VPC dashboard' lists various VPC-related services. In the main content area, a success message 'You successfully created vpc-0959c287edca344af / test-vpc-sagar' is displayed above the 'vpc-0959c287edca344af / test-vpc-sagar' page. The page displays detailed information about the VPC, including its ID, state, tenancy, default VPC, and network metrics. Below the details, there are tabs for 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. At the bottom right, there is an 'Activate Windows' watermark.

Subnet Creation

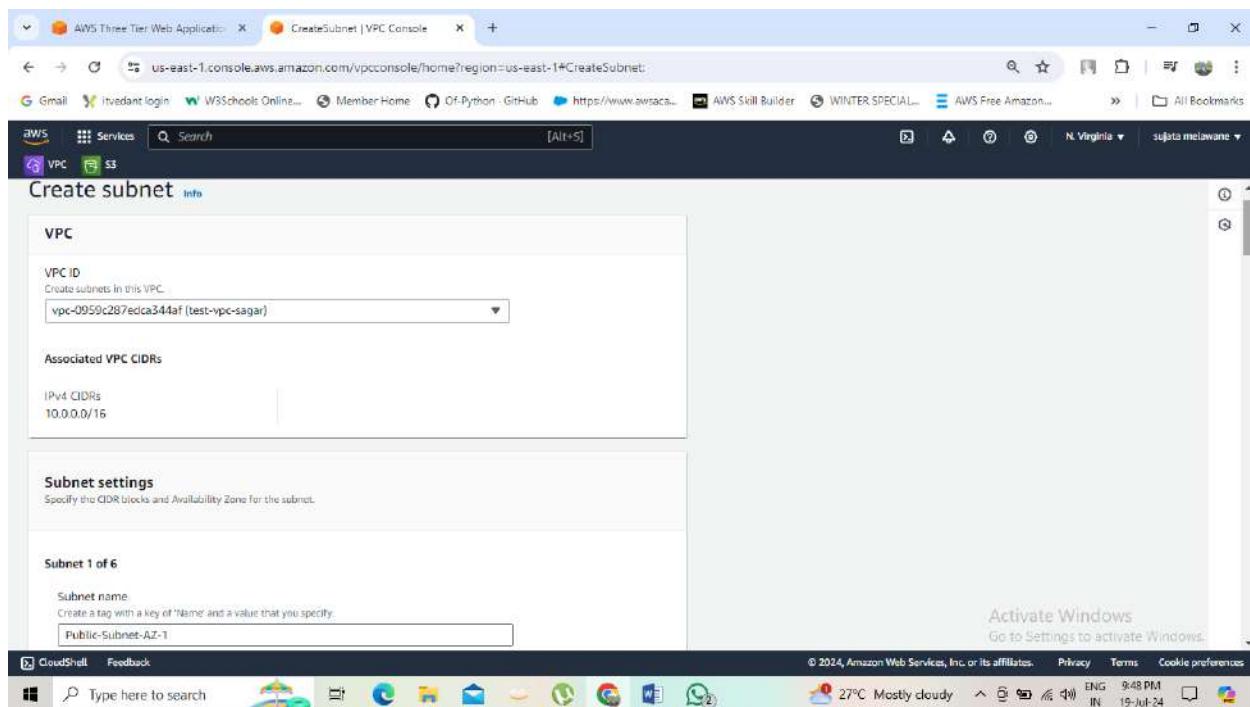
1. Next, create your subnets by navigating to **Subnets** on the left side of the dashboard and clicking **Create subnet**.

The screenshot shows the AWS VPC console interface, specifically the 'Subnets' section. On the left, the 'VPC dashboard' sidebar is visible. The main area displays a table titled 'Subnets (5) info' showing five successfully created subnets. The table includes columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. Each subnet is listed with its unique identifier and configuration. Below the table, there is a search bar and a link to 'Create subnet'. At the bottom right, there is an 'Activate Windows' watermark.

2. We will need **six** subnets across **two** availability zones. That means that **three** subnets will be in one availability zone, and three subnets will be in another zone. Each subnet in one availability zone will correspond to one layer of our three tier architecture. Create each of the 6 subnets by specifying the VPC we created in part 1 and then choose a name, availability zone, and appropriate CIDR range for each of the subnets.

*NOTE: It may be helpful to have a naming convention that will help you remember what each subnet is for. For example in one AZ you might have the following: **Public-Web-Subnet-AZ-1**, **Private-App-Subnet-AZ-1**, **Private-DB-Subnet-AZ-1**.*

NOTE: Remember, your CIDR range for the subnets will be subsets of your VPC CIDR range.



AWS Three Tier Web Application CreateSubnet | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

aws Services Search [Alt+S]

VPC Subnet settings

Specify the CIDR blocks and Availability zone for the subnet.

Subnet 1 of 6

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Public-Subnet-AZ-1"/>

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 9:49 PM IN 19-Jul-24

Type here to search

AWS Three Tier Web Application CreateSubnet | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

aws Services Search [Alt+S]

VPC Subnet 2 of 6

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="Private-Subnet-AZ-1"/>

Add new tag

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 9:49 PM IN 19-Jul-24

Type here to search

AWS Three Tier Web Application

CreateSubnet | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

Subnet 3 of 6

Subnet name Create a tag with a key of 'Name' and a value that you specify.

Private-DB-Subnet-AZ-1

The name can be up to 256 characters long.

Availability Zone Info Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1b

IPv4 VPC CIDR block Info Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.0.0.0/16

IPv4 subnet CIDR block

10.0.2.0/24 256 IPs

Tags - optional

Add new tag You can add up to 40 more tags.

Key Value - optional

Name Private-DB-Subnet-AZ-1 Remove

Add new tag

CloudShell Feedback Type here to search

Activate Windows Go to Settings to activate Windows.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 9:49 PM IN 19-Jul-24

AWS Three Tier Web Application

CreateSubnet | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

Subnet 4 of 6

Subnet name Create a tag with a key of 'Name' and a value that you specify.

Public-Subnet-AZ-2

The name can be up to 256 characters long.

Availability Zone Info Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1c

IPv4 VPC CIDR block Info Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.0.0.0/16

IPv4 subnet CIDR block

10.0.3.0/24 256 IPs

Tags - optional

Add new tag You can add up to 40 more tags.

Key Value - optional

Name Public-Subnet-AZ-2 Remove

Add new tag

CloudShell Feedback Type here to search

Activate Windows Go to Settings to activate Windows.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 9:50 PM IN 19-Jul-24

AWS Three Tier Web Application CreateSubnet | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet

G Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

aws Services Search [Alt+S]

VPC S3

Subnet 5 of 6

Subnet name Create a tag with a key of 'Name' and a value that you specify. Private-Subnet-AZ-2 The name can be up to 256 characters long.

Availability Zone Info Choose the zone in which your subnet will reside, or let Amazon choose one for you. US East (N. Virginia) / us-east-1c

IPv4 VPC CIDR block Info Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block. 10.0.0.0/16

IPv4 subnet CIDR block 10.0.4.0/24 256 IPs

Tags - optional

Add new tag You can add up to 40 more tags.

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Mostly cloudy ENG 9:50 PM IN 19-Jul-24

Activate Windows Go to Settings to activate Windows.

AWS Three Tier Web Application CreateSubnet | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet

G Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

aws Services Search [Alt+S]

VPC S3

Subnet 6 of 6

Subnet name Create a tag with a key of 'Name' and a value that you specify. Private-DB-Subnet-AZ-2 The name can be up to 256 characters long.

Availability Zone Info Choose the zone in which your subnet will reside, or let Amazon choose one for you. US East (N. Virginia) / us-east-1c

IPv4 VPC CIDR block Info Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block. 10.0.0.0/16

IPv4 subnet CIDR block 10.0.5.0/24 256 IPs

Tags - optional

Add new tag You can add up to 40 more tags.

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Mostly cloudy ENG 9:50 PM IN 19-Jul-24

Activate Windows Go to Settings to activate Windows.

Name	Subnet ID	State	VPC	IPv4 CIDR
Public-Subnet-AZ-1	subnet-0bffffd13ded9df9e	Available	vpc-0959c287edca344af test...	10.0.0.0/24
Private-Subnet-AZ-1	subnet-042c8a8c339609bf	Available	vpc-0959c287edca344af test...	10.0.1.0/24
Private-DB-Subnet-AZ-1	subnet-08b2785be7a9bcb47	Available	vpc-0959c287edca344af test...	10.0.2.0/24
Public-Subnet-AZ-2	subnet-071d8ea824469aaf	Available	vpc-0959c287edca344af test...	10.0.3.0/24
Private-Subnet-AZ-2	subnet-023f5eeec8aa69fb01	Available	vpc-0959c287edca344af test...	10.0.4.0/24
Private-DB-Subnet-AZ-2	subnet-032e9828d78704c4d	Available	vpc-0959c287edca344af test...	10.0.5.0/24

Your final subnet setup should be similar to this. Verify that you have 3 subnets across 2 different availability zones.

Internet Connectivity

1. Internet Gateway

- In order to give the public subnets in our VPC internet access we will have to create and attach an Internet Gateway. On the left hand side of the VPC dashboard, select **Internet Gateway**

The screenshot shows the AWS VPC console with the 'Internet gateways' section selected. A table lists one internet gateway:

Name	Internet gateway ID	State	VPC ID	Owner
	igw-01b8ef5007ec3f0e0	Attached	vpc-0f53d8ff0d791abbc4	471112666878

Below the table, a message says 'Select an internet gateway above'. The browser's address bar shows the URL: us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#igws.

Create your internet gateway by simply giving it a name and clicking **Create internet gateway**

The screenshot shows the 'Create internet gateway' wizard. In the 'Internet gateway settings' step, a 'Name tag' field contains 'Three-Tier-igw-sagar'. Below it, a 'Tags - optional' section shows a single tag: 'Name' with value 'Three-Tier-igw-sagar'. At the bottom right is a 'Create internet gateway' button.

After creating the internet gateway, attach it to your VPC that you create in the **VPC** and **Subnet Creation** step of the workshop. You have a couple options on how to do this, either with the creation success message or the **Actions** drop down.

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A message at the top states: 'The following Internet gateway was created: igw-04471280dbada96c7 - Three-Tier-igw-sagar. You can now attach to a VPC to enable the VPC to communicate with the internet.' Below this, the 'igw-04471280dbada96c7 / Three-Tier-igw-sagar' Internet gateway is listed with its details: Internet gateway ID (igw-04471280dbada96c7), State (Detached), VPC ID (-), and Owner (4711126868). The 'Actions' menu includes options like 'Attach to VPC', 'Detach from VPC', 'Manage tags', and 'Delete'. On the left sidebar, under 'Virtual private cloud', the 'Internet gateways' option is also selected. The Windows taskbar at the bottom shows the date as 19-Jul-24 and the time as 10:00 PM.

Then, select the correct VPC and click **Attach internet gateway**.

The screenshot shows the 'Attach to VPC' dialog box for the Internet gateway 'igw-04471280dbada96c7'. The dialog title is 'Attach to VPC (igw-04471280dbada96c7)'. It contains a list of 'Available VPCs' with one item selected: 'vpc-0959c287edca344af - test-vpc-sagar'. At the bottom right of the dialog is a yellow 'Attach internet gateway' button. The Windows taskbar at the bottom shows the date as 19-Jul-24 and the time as 10:01 PM.

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A success message at the top states: "Internet gateway igw-04471280dbada96c7 successfully attached to vpc-0959c287edca344af". Below this, the details for the Internet gateway are displayed:

Internet gateway ID	igw-04471280dbada96c7	State	Attached
VPC ID	vpc-0959c287edca344af test-vpc-sagar	Owner	471112686878

The 'Tags' section shows a single tag: Name = Three-Tier-igw-sagar.

At the bottom of the window, there is a watermark: "Activate Windows Go to Settings to activate Windows."

NAT Gateway

1. In order for our instances in the app layer private subnet to be able to access the internet they will need to go through a NAT Gateway. For high availability, you'll deploy one NAT gateway in each of your **public** subnets. Navigate to **NAT Gateways** on the left side of the current dashboard and click **Create NAT Gateway**.

The screenshot shows the AWS VPC console with the 'NAT gateways' section selected. The left sidebar lists various VPC-related options like 'Your VPCs', 'Subnets', and 'Route tables'. The main pane displays a table titled 'NAT gateways' with a single row: 'No NAT gateways found'. Below the table, a message says 'Select a NAT gateway'. The bottom status bar indicates it's 27°C, mostly cloudy, and shows the date as 19-Jul-24.

Fill in the **Name**, choose one of the **public subnets** you created in part 2, and then allocate an Elastic IP. Click **Create NAT gateway**

The screenshot shows the 'CreateNatGateway' wizard. The first step, 'NAT gateway settings', is displayed. It requires filling in the 'Name' field (with a note about creating a tag), selecting a 'Subnet' (a public subnet is chosen), and choosing a 'Connectivity type' (Public is selected). An 'Elastic IP allocation ID' dropdown is also present. A green success message at the top states 'Elastic IP address 18.211.125.160 (eipalloc-00894be6d630ce1e1) allocated.' The bottom status bar shows it's 27°C, mostly cloudy, and the date is 19-Jul-24.

The screenshot shows the AWS VPC dashboard. A green success message at the top says "NAT gateway nat-0777bca3bb0b46951 | NAT-GW-AZ-1 was created successfully." Below it, the "nat-0777bca3bb0b46951 / NAT-GW-AZ-1" page is displayed. The "Details" section shows the following information:

NAT gateway ID	Connectivity type	State	State message
nat-0777bca3bb0b46951	Public	Pending	Info
NAT gateway ARN	arn:aws:ec2:us-east-1:471112686878:natgateway/nat-0777bca3bb0b46951	Primary public IPv4 address	Primary private IPv4 address
VPC	vpc-0959c287edce344af / test-vpc-sagar	Subnet	Created: Friday, July 19, 2024 at 22:09:11 GMT+5:30

Below the details, there are tabs for "Secondary IPv4 addresses", "Monitoring", and "Tags". The status bar at the bottom indicates "27°C Mostly cloudy" and the date "19-Jul-24".

3. Repeat step 1 and 2 for the other subnet.

The screenshot shows the AWS VPC console. A green success message at the top says "Elastic IP address 34.196.106.221 (eipalloc-0a1fb25be933d19f) allocated." Below it, the "CreateNatGateway" page is displayed. The "NAT gateway settings" section includes the following fields:

- Name - optional:** NAT-GW-AZ-2
- Subnet:** subnet-071d8ea324469aef (Public-Subnet-AZ-2)
- Connectivity type:** Public (radio button selected)
- Elastic IP allocation ID:** eipalloc-0a1fb25be933d19f

At the bottom, there is a link "Additional settings" and the status bar at the bottom indicates "27°C Mostly cloudy" and the date "19-Jul-24".

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Virtual private cloud', 'Route tables' is selected. In the main content area, a success message 'NAT gateway nat-04ae6c38a2183346 | NAT-GW-AZ-2 was created successfully.' is displayed above a table. The table details for the NAT gateway include:

NAT gateway ID	Connectivity type	State	State message
nat-04ae6c38a2183346	Public	Pending	Info
NAT gateway ARN	arn:aws:ec2:us-east-1:47111268678:natgateway/nat-04ae6c38a2183346	Primary public IPv4 address	Primary network interface ID
VPC	vpc-0959c287edce344af / test-vpc-sagar	Subnet	Created: Friday, July 19, 2024 at 22:11:19 GMT+5:30

Below the table are tabs for 'Secondary IPv4 addresses', 'Monitoring', and 'Tags'. The status bar at the bottom shows '27°C Mostly cloudy'.

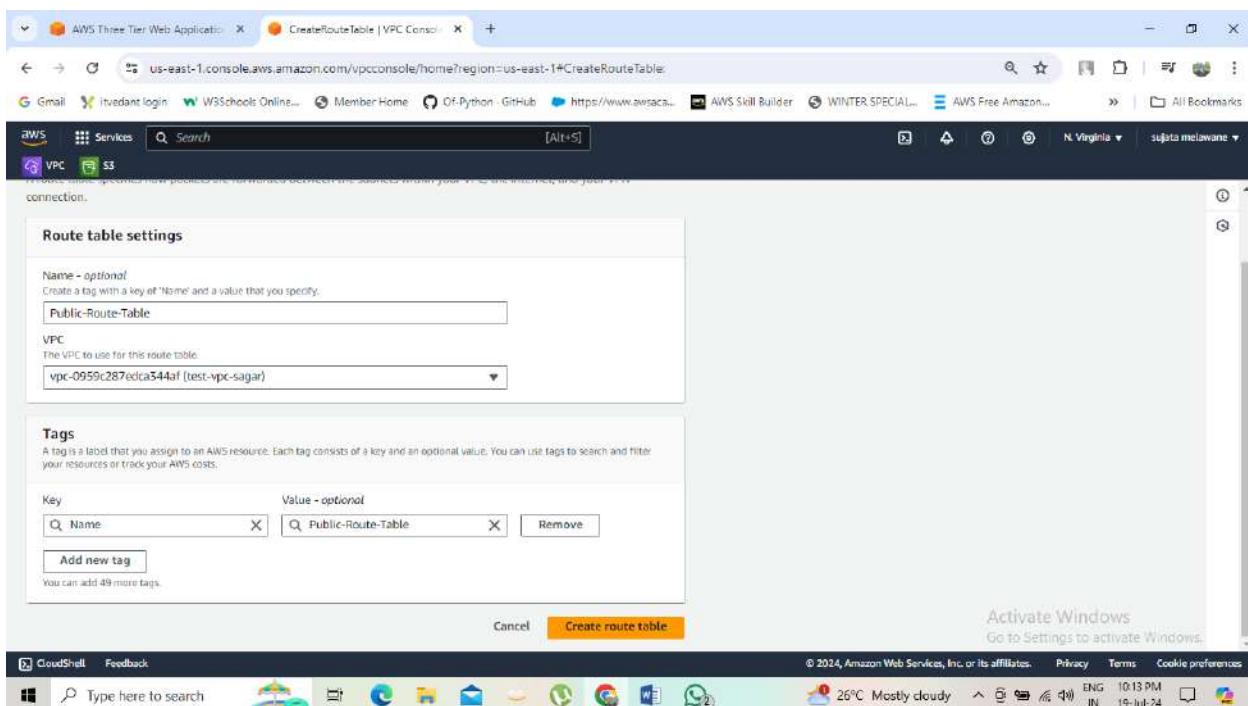
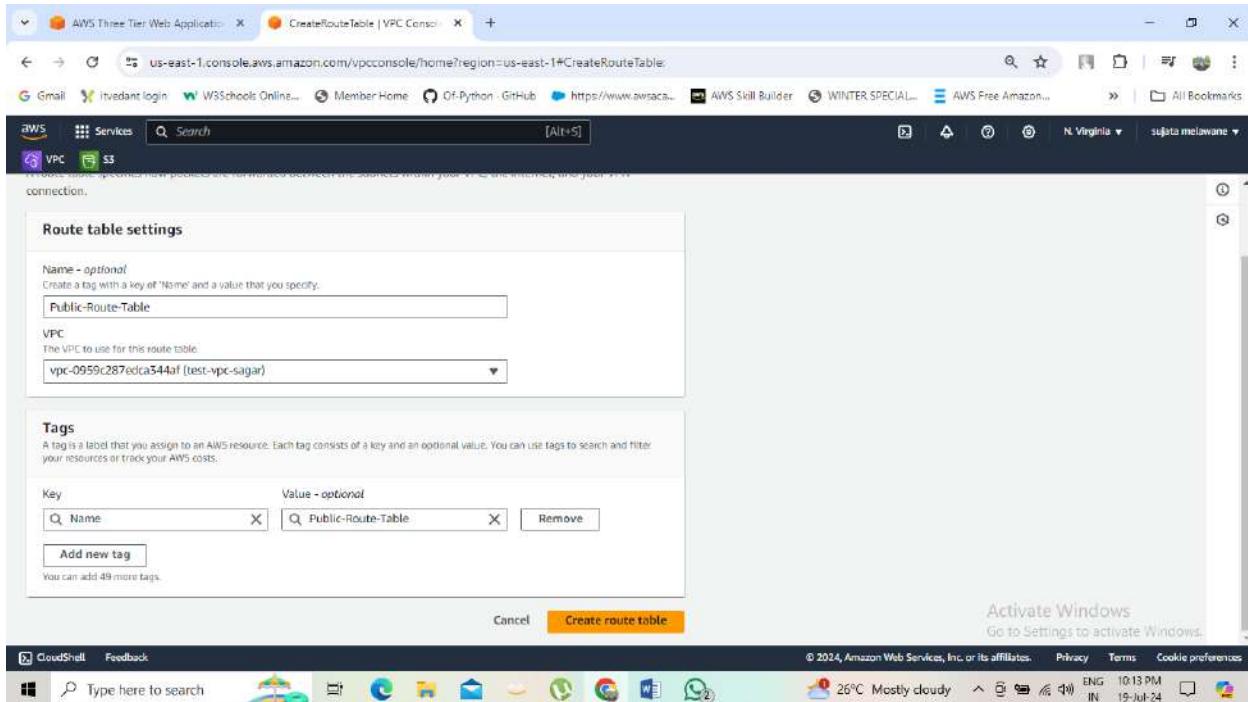
Routing Configuration

1. Navigate to **Route Tables** on the left side of the VPC dashboard and click **Create route table**. First, let's create one route table for the web layer *public subnets* and name it accordingly.

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Virtual private cloud', 'Route tables' is selected. In the main content area, a success message 'NAT gateway nat-04ae6c38a2183346 | NAT-GW-AZ-2 was created successfully.' is displayed above a table. The table details for the route table include:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0dc62fa8195f5cc8e	-	-	Yes	vpc-065dbcf0d791abbc4

Below the table is a section titled 'Select a route table' with three icons: a magnifying glass, a plus sign, and a minus sign. The status bar at the bottom shows '26°C Mostly cloudy'.



2. After creating the route table, you'll automatically be taken to the details page.
Scroll down and click on the **Routes tab** and **Edit routes**.

The screenshot shows the AWS VPC console interface. A success message at the top states "Route table rtb-06f3a26d27af41310 | Public-Route-Table was created successfully." Below this, the "rtb-06f3a26d27af41310 / Public-Route-Table" details are displayed. The "Details" tab is selected, showing the following information:

Route table ID	rtb-06f3a26d27af41310	Main	No	Explicit subnet associations	-	Edge associations	-
VPC	vpc-0959c287edca344af test-vpc-sagar	Owner ID	471112686878				

Below the table, there are tabs for "Routes", "Subnet associations", "Edge associations", "Route propagation", and "Tags". The "Routes" tab is selected, showing one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	Activate Windows No Go to Settings to activate Windows.

The status bar at the bottom indicates "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

The screenshot shows the AWS VPC console interface. A success message at the top states "Route table rtb-06f3a26d27af41310 | Public-Route-Table was created successfully." Below this, the "Route tables (1/3) Info" section is displayed. The table lists three route tables:

Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
rtb-0dc62fa8195f6cc8e	-	-	-	Yes	vpc-053ribif0d791abbcd
Public-Route-Table	rtb-06f3a26d27af41310	-	-	No	vpc-0959c287edca344af
-	rtb-0870d98c581559f6b	-	-	Yes	vpc-0959c287edca344af

Below the table, there are tabs for "Details", "Routes", "Subnet associations", "Edge associations", "Route propagation", and "Tags". The "Routes" tab is selected, showing one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	Activate Windows No Go to Settings to activate Windows.

The status bar at the bottom indicates "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

3. Add a route that directs traffic from the VPC to the internet gateway. In other words, for all traffic *destined* for IPs outside the VPC CIDR range, add an entry that directs it to the internet gateway as a *target*. Save the changes.

The screenshot shows the AWS VPC Edit Routes interface. A new route is being added to a route table. The destination is 0.0.0.0/0, and the target is set to 'Internet Gateway' with the identifier 'igw-04471280dbada96c7'. The status is 'Active' and propagation is 'No'. The 'Save changes' button is highlighted.

The screenshot shows the AWS VPC Route Table Details interface for the route table 'rtb-06f3a26d27af41310'. It displays the route table's details, including its ID, VPC, and owner. Under the 'Routes' tab, two routes are listed: one to 'igw-04471280dbada96c7' and another to 'local'.

4. Edit the **Explicit Subnet Associations** of the route table by navigating to the route table details again. Select **Subnet Associations** and click **Edit subnet associations**.

AWS Three Tier Web Application | RouteTableDetails | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-06f3a26d27af41310

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC dashboard X

EC2 Global View Filter by VPC

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways CloudShell Feedback

Type here to search 26°C Mostly cloudy 10:18 PM ENG 19-Jul-24

VPC Route tables rtb-06f3a26d27af41310 / Public-Route-Table

Actions

Details Info

Route table ID rtb-06f3a26d27af41310	Main No	Explicit subnet associations	Edge associations
VPC vpc-0959e287edca544af test-vpc-sagar	Owner ID 471112666878		

Routes Subnet associations Edge associations Route propagation Tags

Explicit subnet associations (0) Edit subnet associations

No subnet associations You do not have any subnet associations.

Activate Windows Go to Settings to activate Windows.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Three Tier Web Application | EditRouteTableSubnetAssociations

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-06f3a26d27af41310

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

CloudShell Feedback

Type here to search 26°C Mostly cloudy 10:19 PM ENG 19-Jul-24

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/6)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Private-Subnet-AZ-2	subnet-023f5ec&aa69fb01	10.0.4.0/24	-	Main (rtb-0870d98c581559fb)
Private-DB-Subnet-AZ-2	subnet-032e9828d78704cd	10.0.5.0/24	-	Main (rtb-0870d98c581559fb)
Private-Subnet-AZ-1	subnet-042c8a8c339609bf	10.0.1.0/24	-	Main (rtb-0870d98c581559fb)
<input checked="" type="checkbox"/> Public-Subnet-AZ-2	subnet-071d8ea024469aaf	10.0.3.0/24	-	Main (rtb-0870d98c581559fb)
<input type="checkbox"/> Private-DB-Subnet-AZ-1	subnet-08b2785be7a9bcb47	10.0.2.0/24	-	Main (rtb-0870d98c581559fb)
<input checked="" type="checkbox"/> Public-Subnet-AZ-1	subnet-0bfffafdf13ded9df9e	10.0.0.0/24	-	Main (rtb-0870d98c581559fb)

Selected subnets

subnet-071d8ea024469aaf / Public-Subnet-AZ-2 X subnet-0bfffafdf13ded9df9e / Public-Subnet-AZ-1 X

Activate Windows Go to Settings to activate Windows.

Cancel Save associations

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Three Tier Web Application | RouteTableDetails | VPC Console

You have successfully updated subnet associations for rtb-06f3a26d27af41310 / Public-Route-Table.

rtb-06f3a26d27af41310 / Public-Route-Table

Actions

Details **Info**

Route table ID rtb-06f3a26d27af41310	Main No	Explicit subnet associations <u>2 subnets</u>	Edge associations -
VPC vpc-0959c287edca344ef test-vpc-sagar	Owner ID 471112686878		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-04471280dbada96c7	Active	No
10.0.0.16	local	Active	No

Both Edit routes < 1 >

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 10:20 PM IN 19-Jul-24

Type here to search

AWS Three Tier Web Application | RouteTableDetails | VPC Console

Route table rtb-0bb36b4c0ca08be08 | Private-RT-AZ-1 was created successfully.

rtb-0bb36b4c0ca08be08 / Private-RT-AZ-1

Actions

Details **Info**

Route table ID rtb-0bb36b4c0ca08be08	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-0959c287edca344ef test-vpc-sagar	Owner ID 471112686878		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (1)

Destination	Target	Status	Propagated
10.0.0.16	local	Active	No

Both Edit routes < 1 >

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 10:25 PM IN 19-Jul-24

Type here to search

The screenshot shows the 'Edit routes' page for a specific route table. The table lists routes with their destination, target, status, and propagated status. A search bar at the top is set to 'Route tables'.

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No
	nat-0777bca3bb0b46951	-	
	nat-04ae6c38a2183346 (NAT-GW-AZ-2)	-	
	nat-0777bca3bb0b46951 (NAT-GW-AZ-1)	-	

The screenshot shows the VPC dashboard with the details of a specific route table. The 'Subnet associations' tab is selected, showing no explicit subnet associations. The left sidebar shows the VPC dashboard navigation menu.

Once the route tables are created and routes added, add the appropriate subnet associations for each of the app layer private subnets.

AWS Three Tier Web Application EditRouteTableSubnetAssociations

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-0bb36b4c0ca08be08

Gmail Itvedant login W3Schools Online... Member Home Of-Python · GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC Route tables rtb-0bb36b4c0ca08be08 Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/6)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Private-Subnet-AZ-2	subnet-025f9ec8aa69f801	10.0.4.0/24	-	Main (rtb-0870d98c581559f6b)
Private-Subnet-AZ-2	subnet-032e9828d78704c4d	10.0.5.0/24	-	Main (rtb-0870d98c581559f6b)
<input checked="" type="checkbox"/> Private-Subnet-AZ-1	subnet-042c8a8c3396098bf	10.0.1.0/24	-	Main (rtb-0870d98c581559f6b)
Public-Subnet-AZ-2	subnet-071d8ea824469aaf	10.0.3.0/24	-	rtb-06f3a26d27af41310 / Public-Rou
Private-DB-Subnet-AZ-1	subnet-08b2795be769bcb47	10.0.2.0/24	-	Main (rtb-0870d98c581559f6b)
Public-Subnet-AZ-1	subnet-0bfffaf13ded9df9e	10.0.0.0/24	-	rtb-06f3a26d27af41310 / Public-Rou

Selected subnets

subnet-042c8a8c3396098bf / Private-Subnet-AZ-1

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 10:28 PM IN 19-Jul-24

Type here to search

AWS Three Tier Web Application CreateRouteTable

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateRouteTable

Gmail Itvedant login W3Schools Online... Member Home Of-Python · GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC CreateRouteTable

Route table settings

Name - optional Create a tag with a key of 'Name' and a value that you specify. Private-RT-AZ-2

VPC The VPC to use for this route table. vpc-0959c287edca544af (test-vpc-sagar)

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
Q Name	Q Private-RT-AZ-2

Add new tag You can add 49 more tags.

Create route table Cancel

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 10:29 PM IN 19-Jul-24

Type here to search

AWS Three Tier Web Application | RouteTableDetails | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails?RouteTableId=rtb-04b0ce093fc7a0c93

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC dashboard X

Route table rtb-04b0ce093fc7a0c93 | Private-RT-AZ-2 was created successfully.

VPC > Route Tables > rtb-04b0ce093fc7a0c93

rtb-04b0ce093fc7a0c93 / Private-RT-AZ-2 Actions

Details Info

Route table ID: rtb-04b0ce093fc7a0c93 Main: No Explicit subnet associations: - Edge associations: -

VPC: vpc-0959c287edca344af | test-vpc-sagar Owner ID: 471112686878

Routes Subnet associations Edge associations Route propagation Tags

Routes (1) Both Edit routes

Filter routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback Type here to search 26°C Mostly cloudy ENG 10:29 PM IN 19-Jul-24

AWS Three Tier Web Application | EditRoutes | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRoutes?RouteTableId=rtb-04b0ce093fc7a0c93

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC > Route tables > rtb-04b0ce093fc7a0c93 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
Q 0.0.0.0/0	NAT Gateway	-	No
Q nat-	nat-04ae6c38a2183346 (NAT-GW-AZ-2)	-	No
Add route	nat-04ae6c38a2183346 (NAT-GW-AZ-2)	-	No
	nat-0777bca5b00f	-	No

Cancel Preview Save changes

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Mostly cloudy ENG 10:30 PM IN 19-Jul-24

AWS Three Tier Web Application | RouteTableDetails | VPC Console

Updated routes for rtb-04b0ce093fc7a0c93 / Private-RT-AZ-2 successfully

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-04b0ce093fc7a0c93	No	-	-
Owner ID	471112656878		
VPC	vpc-0959c257edca544ef test-vpc-sagar		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-04aee6c38a2185346	Active	No
10.0.0.16	local	Active	No

Activate Windows

Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENGLISH 10:30 PM IN 19-Jul-24

Once the route tables are created and routes added, add the appropriate subnet associations for each of the app layer private subnets.

AWS Three Tier Web Application | EditRouteTableSubnetAssociations | VPC Console

Route tables > rtb-04b0ce093fc7a0c93 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1 / 6)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Private-Subnet-AZ-2	subnet-023f5eec8aa69fb01	10.0.4.0/24	-	Main (rtb-0870d98c581559f6b)
Private-DB-Subnet-AZ-2	subnet-032e9828078704c4d	10.0.5.0/24	-	Main (rtb-0870d98c581559f6b)
Private-Subnet-AZ-1	subnet-042c8a8c539609bf	10.0.1.0/24	-	rtb-0bb36b4c0ca08be08 / Private-RT
Public-Subnet-AZ-2	subnet-071d8ea82446baafdf	10.0.3.0/24	-	rtb-06fa26d27af41310 / Public-Rou
Private-DB-Subnet-AZ-1	subnet-08b2785be7a9bcb47	10.0.2.0/24	-	Main (rtb-0870d98c581559f6b)
Public-Subnet-AZ-1	subnet-0bfafffd13dec9df9	10.0.0.0/24	-	rtb-06fa26d27af41310 / Public-Rou

Selected subnets

subnet-023f5eec8aa69fb01 / Private-Subnet-AZ-2

Activate Windows

Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

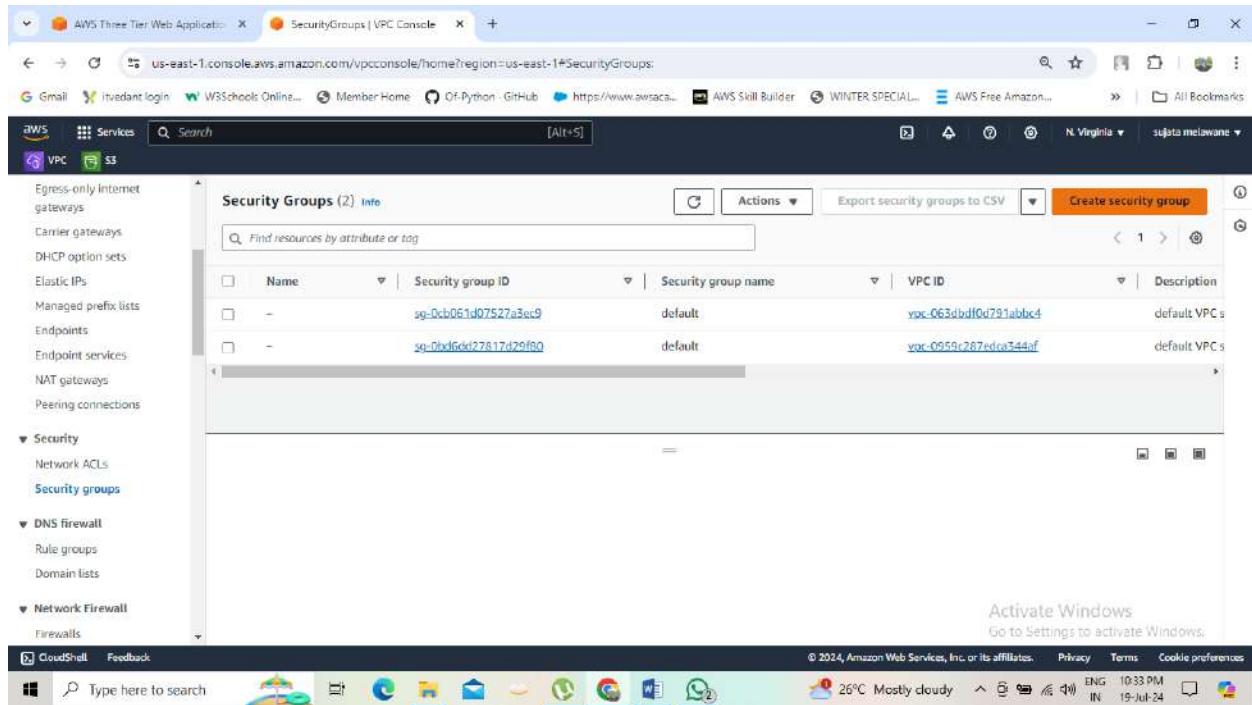
ENGLISH 10:30 PM IN 19-Jul-24

The screenshot shows the AWS VPC console interface. The left sidebar is titled 'VPC dashboard' and includes sections for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, and NAT gateways. The 'Route tables' section is currently selected. The main content area displays a success message: 'You have successfully updated subnet associations for rtb-04b0ce093fc7a0c93 / Private-RT-AZ-2.' Below this, the route table details are shown: Route table ID (rtb-04b0ce093fc7a0c93), Main (No), VPC (vpc-0959c287edca344af | test-vpc-sagar), Owner ID (471112686878), Explicit subnet associations (subnet-023f5ee8aa69f801 / Private-Subnet-AZ-2), and Edge associations (None). A table titled 'Routes (2)' lists two routes: one for destination 0.0.0.0/0 targeting nat-04aee6c38a2183346 (Status: Active, Propagated: No) and another for destination 10.0.0.16 targeting local (Status: Active, Propagated: No). The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time.

4. Now create 2 more route tables, one for each app layer private subnet in each availability zone. These route tables will route app layer traffic destined for outside the VPC to the NAT gateway in the respective availability zone, so add the appropriate routes for that.

Security Groups

1. Security groups will tighten the rules around which traffic will be allowed to our Elastic Load Balancers and EC2 instances. Navigate to **Security Groups** on the left side of the VPC dashboard, under **Security**.



The screenshot shows the AWS VPC Security Groups console. The left sidebar is collapsed, showing options like Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. Under the Security section, Network ACLs and Security groups are listed, with Security groups being the active tab. The main content area displays a table titled "Security Groups (2) Info". The table has columns: Name, Security group ID, Security group name, VPC ID, and Description. It lists two entries: one named "default" with ID "sg_0cb061d07e27a3ec9" and VPC ID "vpc_063dbd10d791abbc4", and another named "default" with ID "sg_d0d5dd27817d29f80" and VPC ID "var_0559c287edca344uf". A search bar at the top of the table says "Find resources by attribute or tag". At the bottom right of the table, there are icons for "Create security group", "Actions", "Export security groups to CSV", and "Create security group". The status bar at the bottom indicates "Activate Windows Go to Settings to activate Windows." and shows system information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences, 26°C Mostly cloudy, ENG 10:33 PM IN 19-Jul-24.

Name	Security group ID	Security group name	VPC ID	Description
-	sg_0cb061d07e27a3ec9	default	vpc_063dbd10d791abbc4	default VPCs
-	sg_d0d5dd27817d29f80	default	var_0559c287edca344uf	default VPCs

2. The first security group you'll create is for the public, **internet facing** load balancer. After typing a name and description, add an inbound rule to allow **HTTP** type traffic for your **IP**.

AWS Three Tier Web Application | CreateSecurityGroup | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSecurityGroup;

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

Basic details

Security group name **Info**
Internet-Facing-LB-SG
Name cannot be edited after creation.

Description **Info**
External-Load-balancer-security-group

VPC Info
vpc-0959c287edca544af (test-vpc-sagar)

Inbound rules **Info**

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	My IP	114.79.138.90/52

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 10:39 PM IN 19-Jul-24

AWS Three Tier Web Application | SecurityGroup | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup;groupId=sg-0f3990664fcfe796

Gmail Itvedant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC dashboard

Details

Security group name Internet-Facing-LB-SG	Security group ID sg-0f3990664fcfe796	Description External-Load balancer-security-group	VPC ID vpc-0959c287edca544af
Owner 471112686878	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules **Outbound rules** **Tags**

Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-02255ae5f423c7fd9	IPv4	HTTP	TCP	80

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 10:41 PM IN 19-Jul-24

3. The second security group you'll create is for the public instances in the web tier. After typing a name and description, add an inbound rule that allows **HTTP** type traffic from your internet facing load balancer security group you created in the previous step. This will allow traffic from your public facing load balancer to hit your instances. Then, add an additional rule that will allow HTTP type traffic for your IP. This will allow you to access your instance when we test.

The screenshot shows the 'CreateSecurityGroup' page in the AWS VPC console. The 'Security group name' field contains 'Web-Tier-SG'. The 'Description' field contains 'SG for the Web Tier'. Under 'VPC info', the 'vpc-0559c287edca544af (test-vpc-sagar)' is selected. The 'Inbound rules' section shows two rules:

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	My IP	114.79.138.90/32
HTTP	TCP	80	Custom	sg-0f3990664fcfe796

At the bottom, there are links for CloudShell, Feedback, and a search bar. The status bar shows '26°C Mostly cloudy' and the date '19-Jul-24'.

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services), and NAT gateway. The main area displays a security group named "Web-Tier-5G" with the following details:

Security group name	sg-06216c771be90b77b	Description	VPC ID
Owner	471112686878	Inbound rules count: 2 Permission entries	vpc-0959c287edca344af
		Outbound rules count: 1 Permission entry	

The "Inbound rules" tab is selected, showing two entries:

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-05a45034f97cec151	-	HTTP	TCP	80
-	sgr-03397f3ce866da68a	IPv4	HTTP	TCP	80

4. The third security group will be for our internal load balancer. Create this new security group and add an inbound rule that allows **HTTP** type traffic from your public instance security group. This will allow traffic from your web tier instances to hit your internal load balancer.

The screenshot shows the "CreateSecurityGroup" wizard. The "Basic details" step is selected. The security group is named "Internet-lb-SG" with the description "SG for internal load balancer". The VPC is set to "vpc-0959c287edca344af (test-vpc-sagar)".

The "Inbound rules" section shows one rule being added:

Type	Protocol	Port range	Source	Action
HTTP	TCP	80	Custom	Add rule

The "Security Groups" dropdown shows the "Web-Tier-5G" security group selected. Other options include "Internet-Facing-LB-SG | sg-0f399064fc" and "default | sg-0bd16dd27817d29f80".

The screenshot shows the AWS VPC Console with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup:groupId=sg-09ecf0544d9acc2f0>. The left sidebar shows the VPC dashboard with various options like EC2 Global View, Virtual private cloud, and Subnets. The main panel displays the 'Details' section for a security group named 'Internet-lb-SG'. It shows the security group ID as sg-09ecf0544d9acc2f0, owner as 471112686878, and a single inbound rule entry. The 'Inbound rules' tab is active, showing the rule 'sg-0f8fd0e182136fb02' which allows HTTP traffic on port 80.

5. The fourth security group we'll configure is for our private instances. After typing a name and description, add an inbound rule that will allow **TCP** type traffic on port **4000** from the **internal load balancer security group** you created in the previous step. This is the port our app tier application is running on and allows our internal load balancer to forward traffic on this port to our private instances. You should also add another route for port **4000** that allows **your IP** for testing.

- The fifth security group we'll configure protects our private database instances. For this security group, add an inbound rule that will allow traffic from the private instance security group to the MySQL/Aurora port (3306).

AWS Three Tier Web Application | CreateSecurityGroup | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSecurityGroup;

Gmail Itivadant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC Security Groups Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
DBSG
Name cannot be edited after creation.

Description Info
SG for our database

VPC Info
vpc-0959c287edca344af (test-vpc-sagar)

Inbound rules Info

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Mostly cloudy ENG IN 10:58 PM 19-Jul-24

AWS Three Tier Web Application | CreateSecurityGroup | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSecurityGroup;

Gmail Itivadant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC Services Search [Alt+S]

VPC Security Groups Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
DBSG
Name cannot be edited after creation.

Description Info
SG for our database

VPC Info
vpc-0959c287edca344af (test-vpc-sagar)

Inbound rules Info

Type Info Protocol Info Port range Info Source Info

MySQL/Aurora TCP 3306 Custom

Add rule

Prefix lists Private-Instance-SG | sg-08eb25389580b547a com.amazonaws.us-east-1.dynamodb | pl-02cd2c6b com.amazonaws.us-east-1.ipv6.route53-healthchecks | pl-05c0959a59362110e

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Mostly cloudy ENG IN 10:59 PM 19-Jul-24

AWS Three Tier Web Application X SecurityGroup | VPC Console X +

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup:groupId=sg-0eb2bfb7e0d7c2859

Gmail ivyendant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks N. Virginia sujata melawane

VPC dashboard EC2 Global View Filter by VPC Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways CloudShell Feedback Type here to search [Alt+S] Details Security group name DBSG Security group ID sg-0eb2bfb7e0d7c2859 Description SG for our database VPC ID vpc-0959c287edca544af Owner 471112686878 Inbound rules count 1 Permission entry Outbound rules count 1 Permission entry Inbound rules Outbound rules Tags Inbound rules (1) Search Manage tags Edit inbound rules Name Security group rule... IP version Type Protocol Port range sg-0acc8e1c9ae073391 - MYSQL/Aurora TCP 3306 Activate Windows Go to Settings to activate Windows. © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 26°C Mostly cloudy ENG 11:00 PM IN 19-Jul-24

The screenshot shows the AWS VPC Console interface. On the left, there's a sidebar with various navigation options like EC2 Global View, Virtual private cloud, and NAT gateways. The main area is titled 'Details' for a security group named 'DBSG'. It shows the security group ID as 'sg-0eb2bfb7e0d7c2859', owned by '471112686878', with one inbound rule for 'MYSQL/Aurora' on 'TCP' port '3306'. The 'Inbound rules' tab is selected. At the bottom right of the page, there's a watermark for 'Activate Windows'.

Part 2: Database Deployment

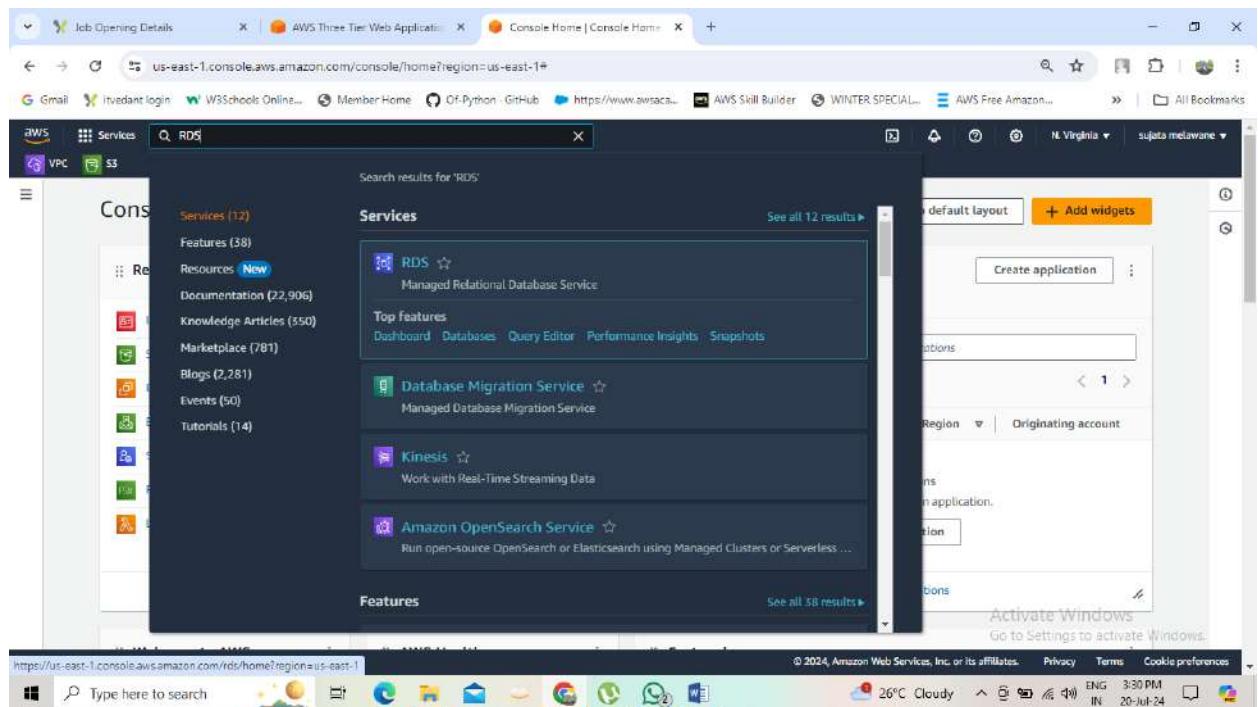
This section of the workshop will walk you through deploying the database layer of the three tier architecture.

Learning Objectives:

- Deploy Database Layer
 - Subnet Groups
 - Multi-AZ Database

Subnet Groups

1. Navigate to the RDS dashboard in the AWS console and click on **Subnet groups** on the left hand side. Click **Create DB subnet group**



The screenshot shows the AWS RDS Subnet Groups page. The left sidebar has 'Subnet groups' selected under 'Services'. The main area shows a table titled 'Subnet groups (0)' with columns for Name, Description, Status, and VPC. A message at the bottom says 'No db subnet groups' and 'You don't have any db subnet groups.' with a 'Create DB subnet group' button.

2. Give your subnet group a name, description, and choose the VPC we created.

The screenshot shows the 'Create DB subnet group' wizard. The 'Subnet group details' step is active. It requires a 'Name' (Three-Tier-db-subnet-group), a 'Description' (subnet group for the database layer of the architecture), and a 'VPC' (test-vpc-sagar). The 'Add subnets' step is shown below.

When adding subnets, make sure to add the subnets we created in each availability zone specifically for our database layer. You may have to navigate back to the VPC dashboard and check to make sure you're selecting the correct subnet IDs

Job Opening Details | AWS Three Tier Web Application | RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#create-db-subnet-group:

Gmail Itivadant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks

AWS Services Search [Alt+S]

VPC SS

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

us-east-1b X us-east-1c X

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

subnet-08b2785be7a9bcb47 (10.0.2.0/24) X
subnet-032e9828d78704c4d (10.0.5.0/24) X

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-08b2785be7a9bcb47	10.0.2.0/24
us-east-1c	subnet-032e9828d78704c4d	10.0.5.0/24

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Cloudy ENG 3:36 PM IN 20-Jul-24

Job Opening Details | AWS Three Tier Web Application | RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#create-db-subnet-group:

Gmail Itivadant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks

AWS Services Search [Alt+S]

VPC SS

us-east-1b X us-east-1c X

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

subnet-08b2785be7a9bcb47 (10.0.2.0/24) X
subnet-032e9828d78704c4d (10.0.5.0/24) X

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

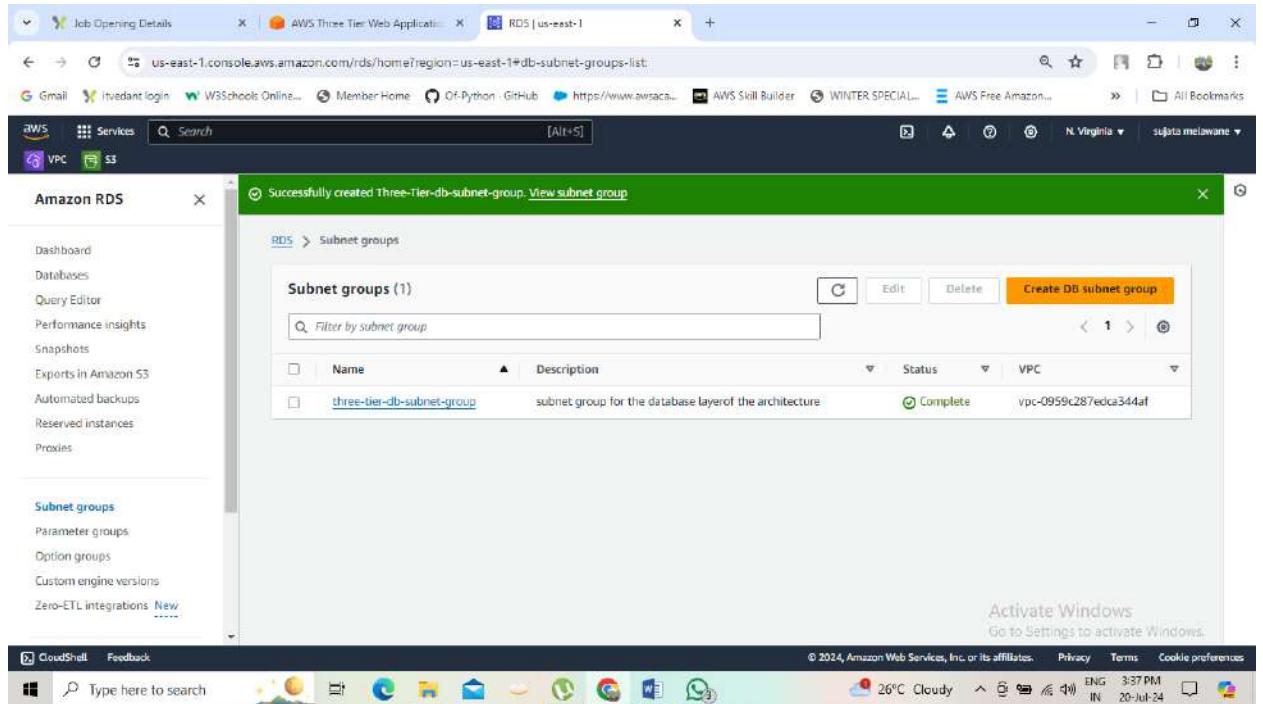
Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-08b2785be7a9bcb47	10.0.2.0/24
us-east-1c	subnet-032e9828d78704c4d	10.0.5.0/24

Create Cancel

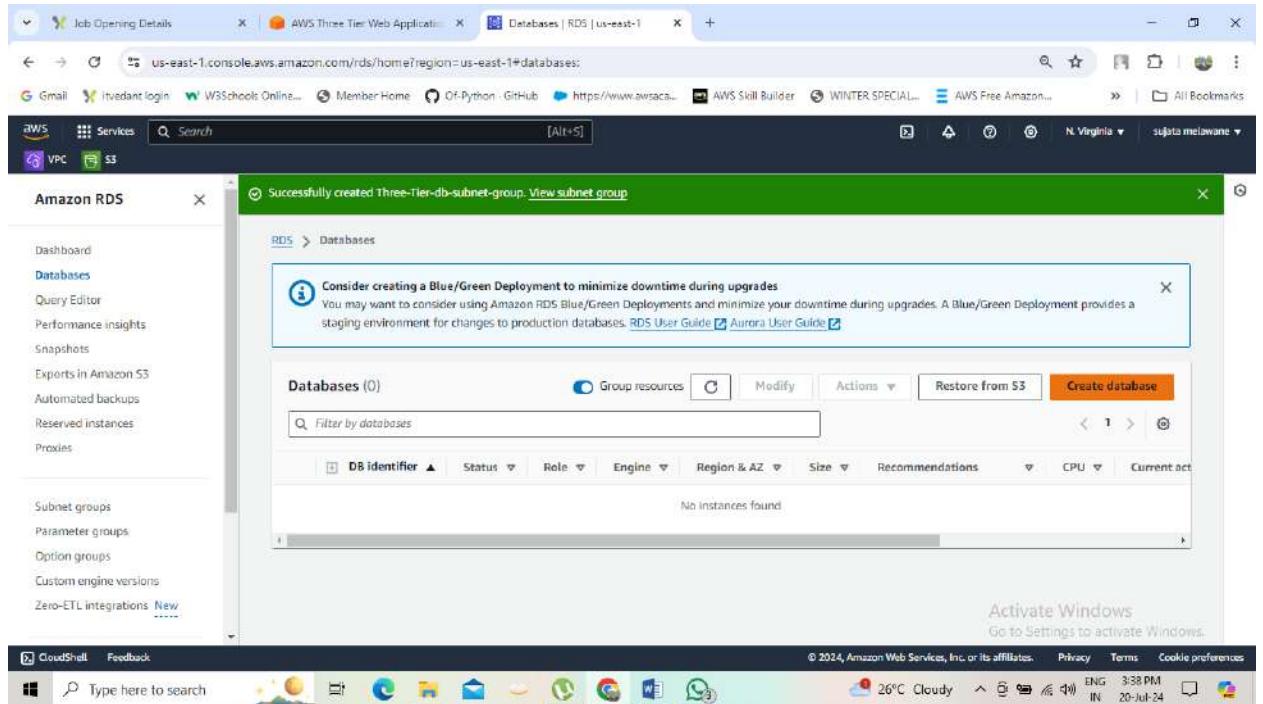
Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Cloudy ENG 3:36 PM IN 20-Jul-24

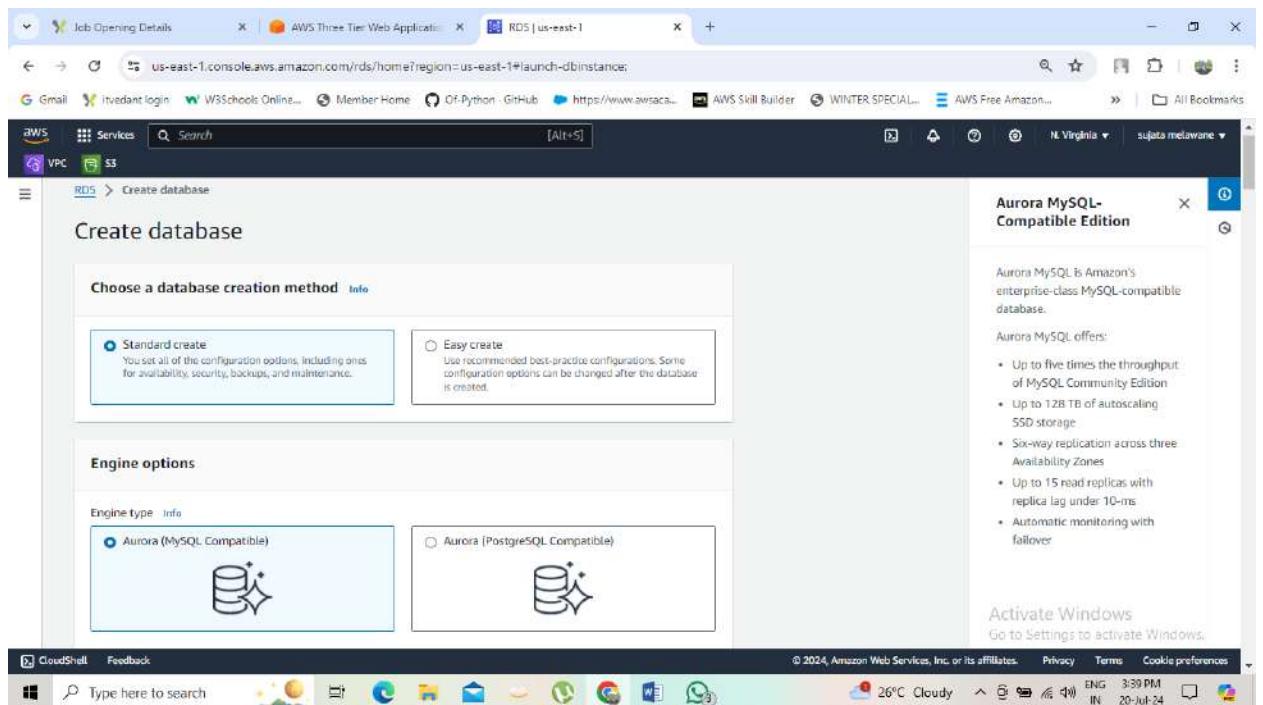


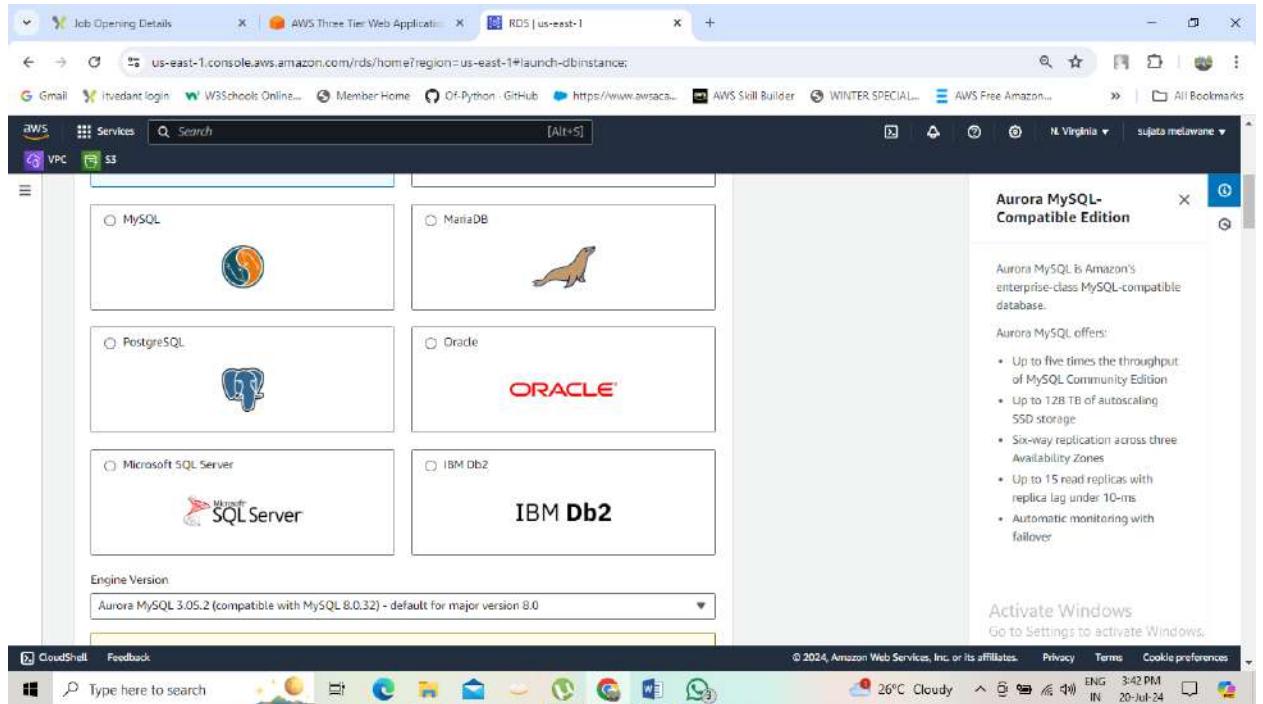
Database Deployment

1. Navigate to **Databases** on the left hand side of the RDS dashboard and click **Create database**.

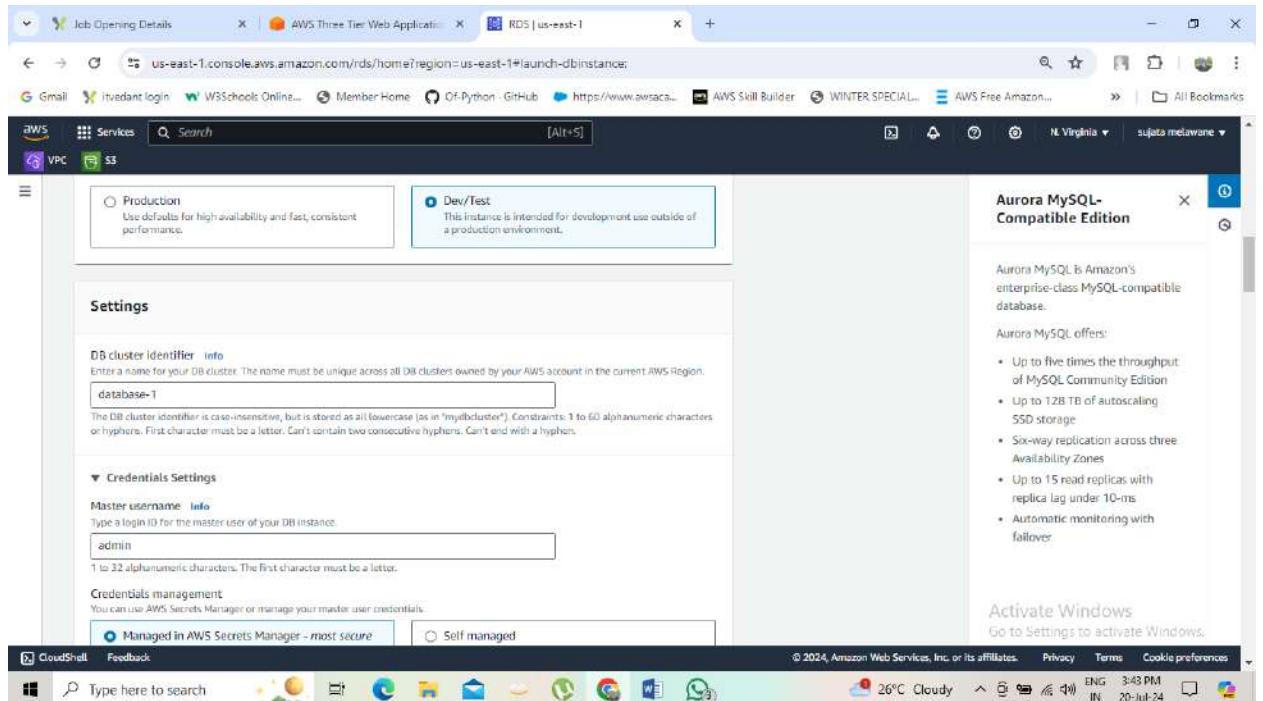


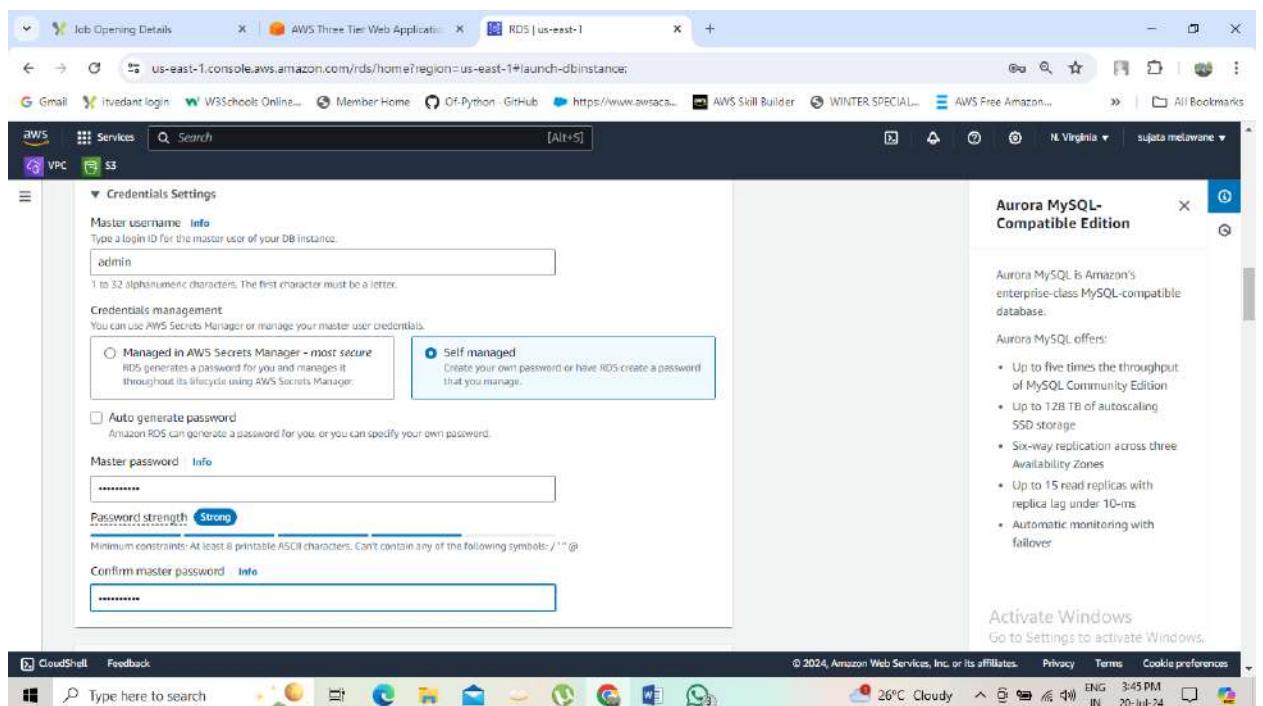
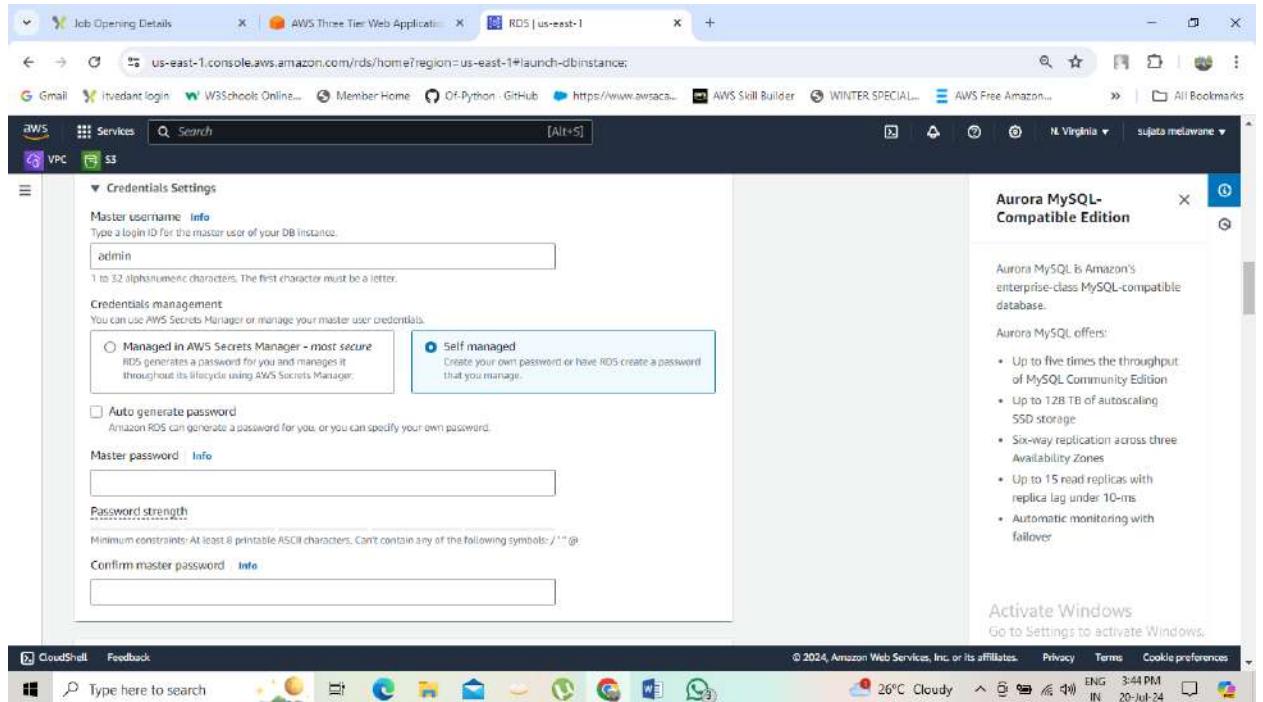
2. We'll now go through several configuration steps. Start with a **Standard create** for this **MySQL-Compatible Amazon Aurora** database. Leave the rest of the defaults in the **Engine options** as default.





Under the **Templates** section choose **Dev/Test** since this isn't being used for production at the moment. Under **Settings** set a username and password of your choice and note them down since we'll be using password authentication to access our database.





PASS-Bhokare11

Next, under **Availability and durability** change the option to create an Aurora Replica or reader node in a different availability zone. Under **Connectivity**, set the VPC, choose the subnet group we created earlier, and select no for public access.

Cluster storage configuration - new Info

Choose the storage configuration for the Aurora DB cluster that best fits your application's price predictability and price performance needs.

Configuration options

Database instance, storage, and I/O charges vary depending on the configuration. [Learn more](#)

Aurora Standard

- Cost-effective pricing for many applications with moderate I/O usage (I/O costs <25% of total database costs).
- Pay-per-request I/O charges apply. DB instance and storage prices don't include I/O usage.

Aurora I/O-Optimized

- Predictable pricing for all applications. Improved price performance for I/O-intensive applications (I/O costs >25% of total database costs).
- No additional charges for read/write I/O operations. DB instance and storage prices include I/O usage.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class Info

Hide filters

Include previous generation classes

Serverless v2

Memory optimized classes (includes r classes)

Aurora MySQL-Compatible Edition

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 1.2B TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-milliseconds
- Automatic monitoring with failover

Job Opening Details X AWS Three Tier Web Application X RDS | us-east-1 X

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance;

G Gmail Y LinkedIn login W W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks

iAWS Services Search [Alt+S]

VPC S

Availability & durability

Multi-AZ deployment [Info](#)

Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)
Creates an Aurora Replica for failover and high availability.

Don't create an Aurora Replica.

Connectivity [Info](#)

C

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.

Network type [Info](#)

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Aurora MySQL-Compatible Edition

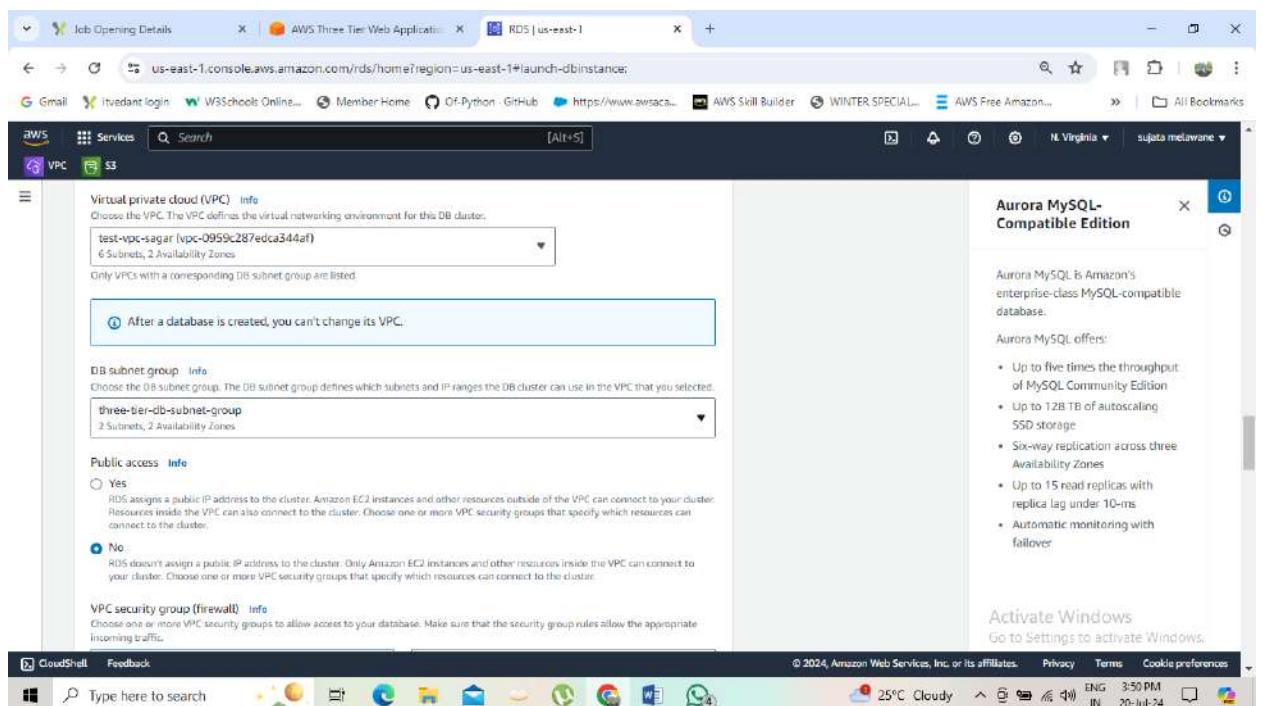
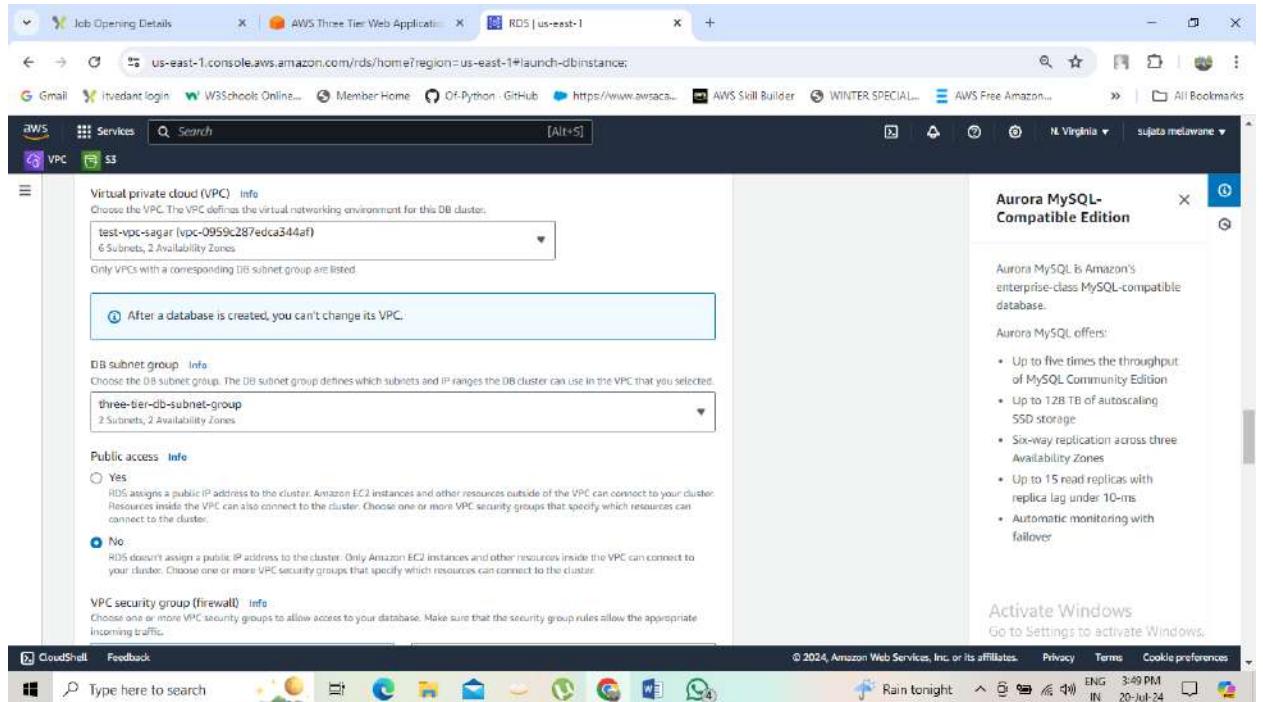
Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

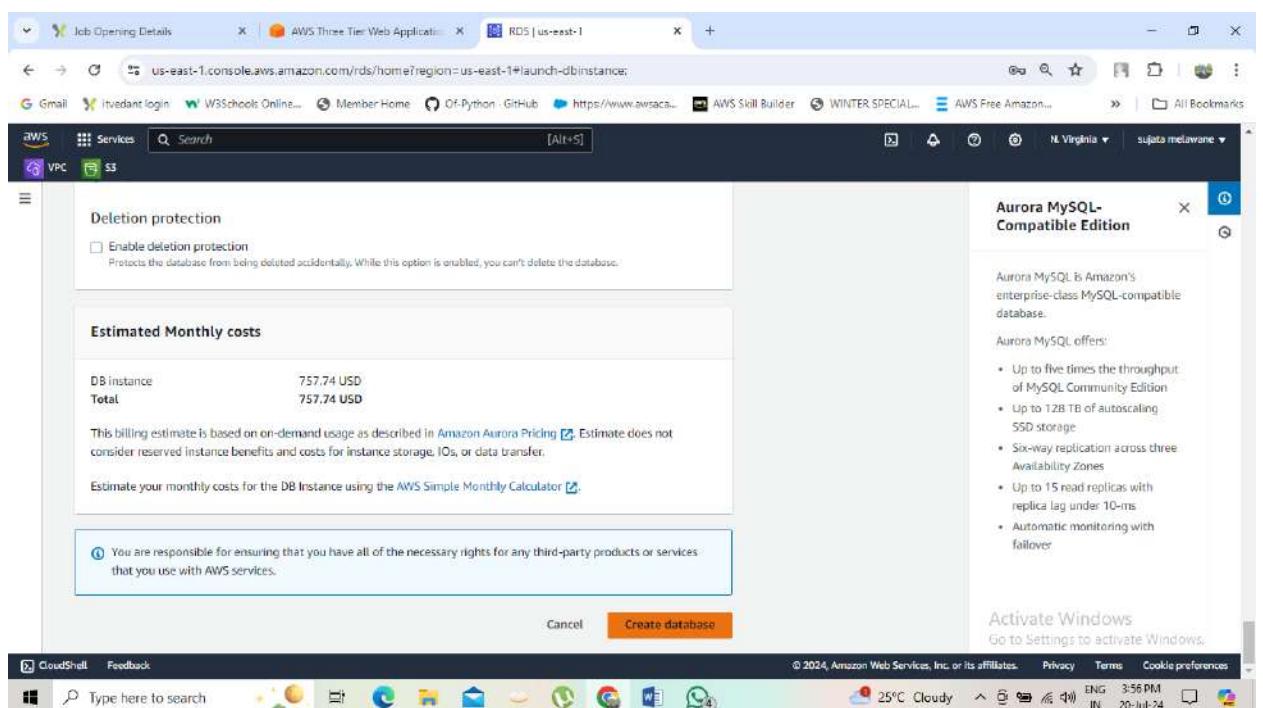
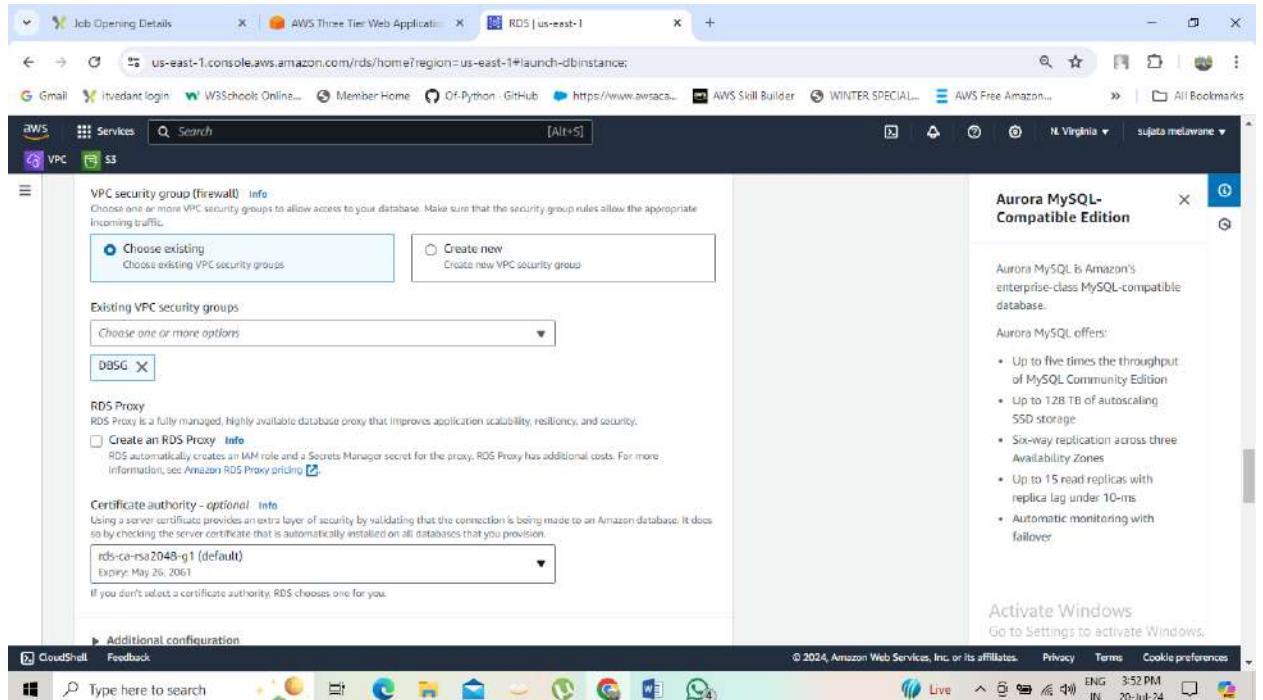
- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Type here to search Rain tonight ENG IN 3:48 PM 30-Nov-24



Set the security group we created for the database layer, make sure **password authentication** is selected as our authentication choice, and create the database.



- When your database is provisioned, you should see a reader and writer instance in the database subnets of each availability zone. Note down the writer endpoint for your database for later use.

Job Opening Details | AWS Three Tier Web Application | Databases | RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases;

Gmail Itivadant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks

AWS Services Search [Alt+S]

Amazon RDS

Dashboard Databases Query Editor Performance insights Snapshots Exports in Amazon S3 Automated backups Reserved instances Proxies Subnet groups Parameter groups Option groups Custom engine versions Zero-ETL integrations New

Successfully created database database-1

You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you.

Consider creating a Blue/Green Deployment to minimize downtime during upgrades You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (3) Group resources Modify Actions Restore from S3 Create database

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Recom
database-1	Available	Regional cluster	Aurora MySQL	us-east-1	2 instances	
database-1-instance-1	Creating	Reader instance	Aurora MySQL	us-east-1b	db.r6g.2xlarge	
database-1-instance-1-us-east-1c	Creating	Reader instance	Aurora MySQL	us-east-1c	db.r6g.2xlarge	Activate Windows

Go to Settings to activate Windows.

CloudShell Feedback Type here to search 25°C Cloudy ENG 3:58 PM IN 20-Jul-24

Job Opening Details | AWS Three Tier Web Application | RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=database-1;is-cluster=true

Gmail Itivadant login W3Schools Online... Member Home Of-Python - GitHub https://www.awacs... AWS Skill Builder WINTER SPECIAL... AWS Free Amazon... All Bookmarks

AWS Services Search [Alt+S]

Amazon RDS

Dashboard Databases Query Editor Performance insights Snapshots Exports in Amazon S3 Automated backups Reserved instances Proxies Subnet groups Parameter groups Option groups Custom engine versions Zero-ETL integrations New

Related

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Recon
database-1	Available	Regional cluster	Aurora MySQL	us-east-1	2 instances	
database-1-instance-1	Modifying	Writer instance	Aurora MySQL	us-east-1b	db.r6g.2xlarge	
database-1-instance-1-us-east-1c	Creating	Reader instance	Aurora MySQL	us-east-1c	db.r6g.2xlarge	

Connectivity & security Monitoring Logs & events Configuration Zero-ETL integrations Maintenance & backups Tags

Endpoints (2)

Find resources Actions Create custom endpoint

Endpoint name	Status	Type	Port
database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com	Available	Writer	3306
database-1.cluster-ro-cj2kcc44ec41.us-east-1.rds.amazonaws.com	Available	Reader	3306

Activate Windows Go to Settings to activate Windows.

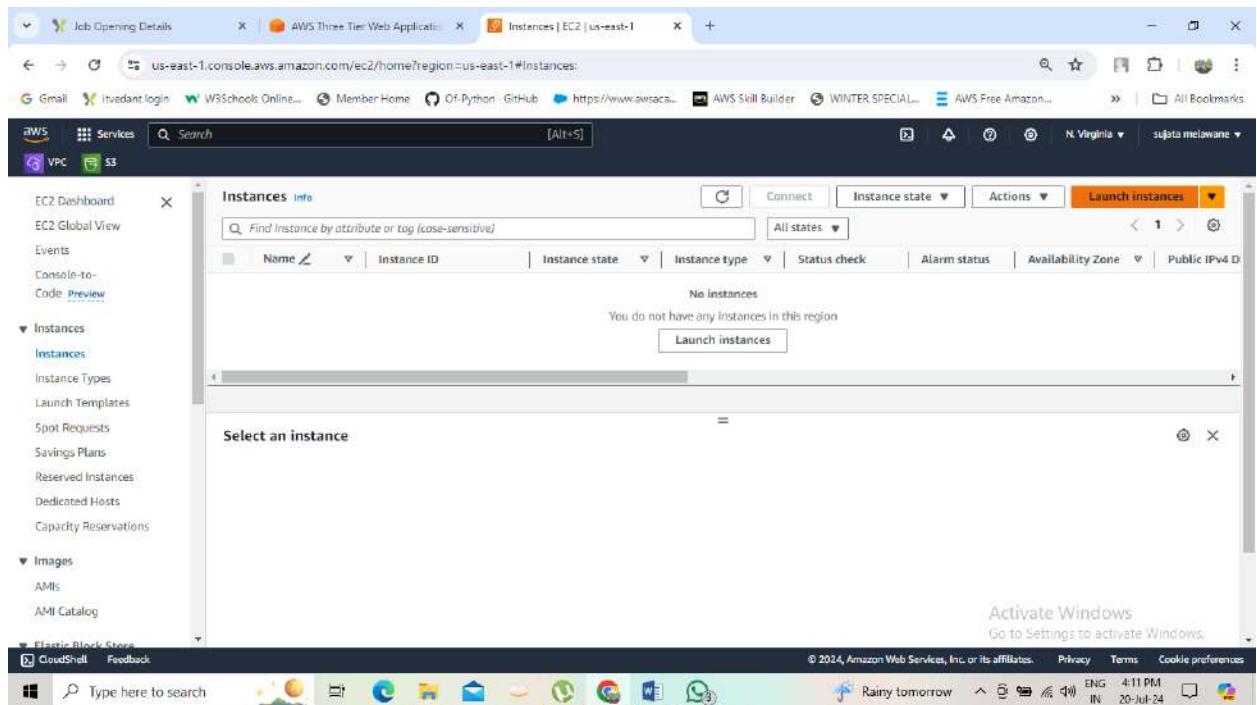
CloudShell Feedback Type here to search 25°C Cloudy ENG 4:03 PM IN 20-Jul-24

Part 3: App Tier Instance Deployment

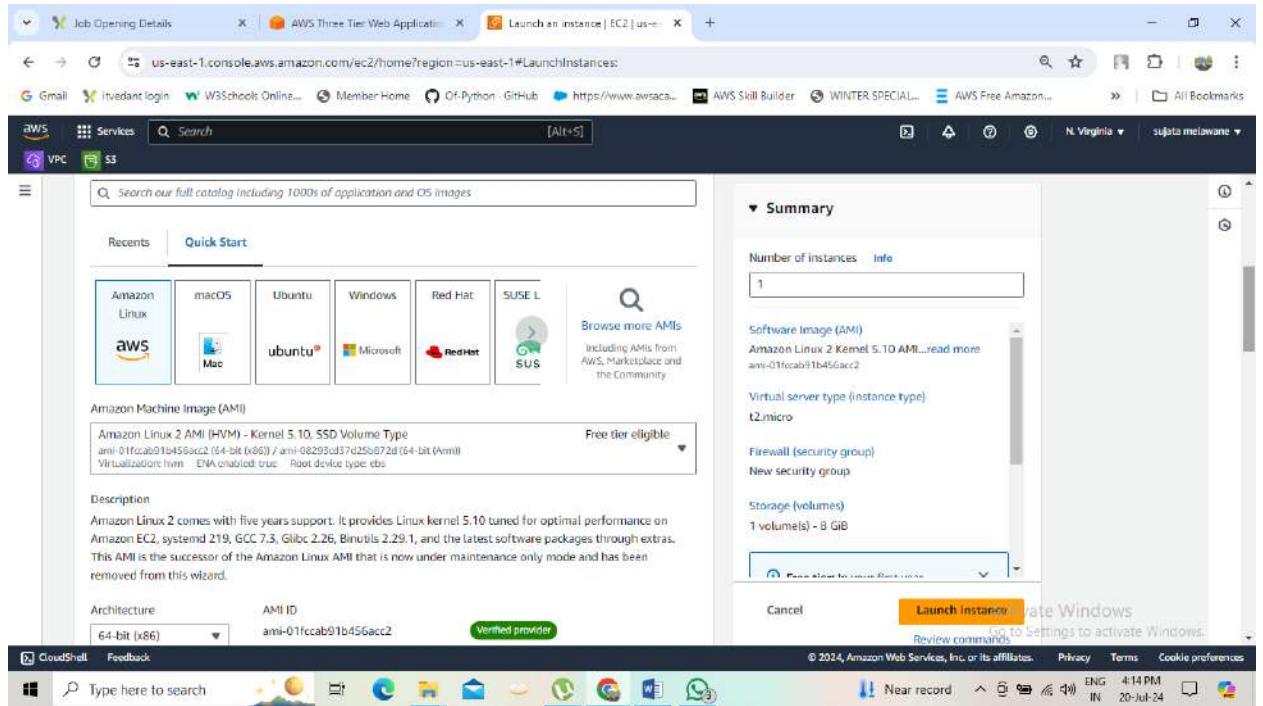
In this section of our workshop we will create an EC2 instance for our app layer and make all necessary software configurations so that the app can run. The app layer consists of a Node.js application that will run on port 4000. We will also configure our database with some data and tables.

Learning Objectives:

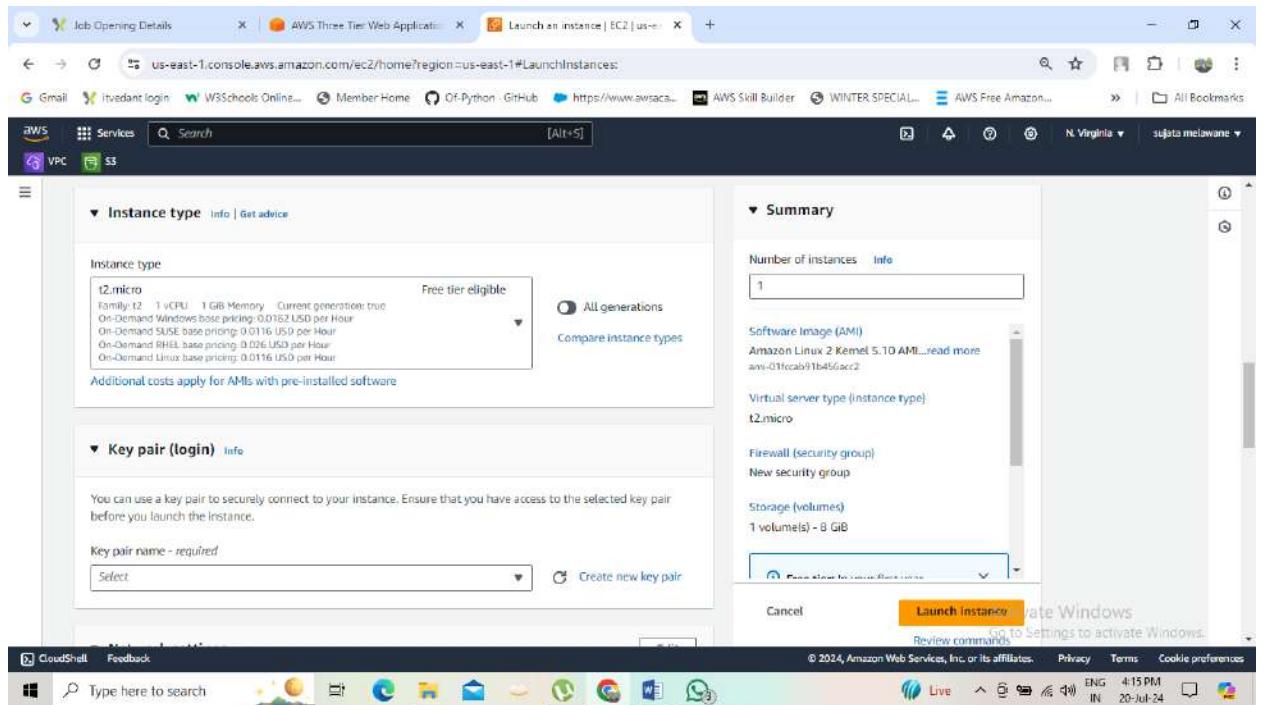
- Create App Tier Instance
- Configure Software Stack
- Configure Database Schema
- Test DB connectivity



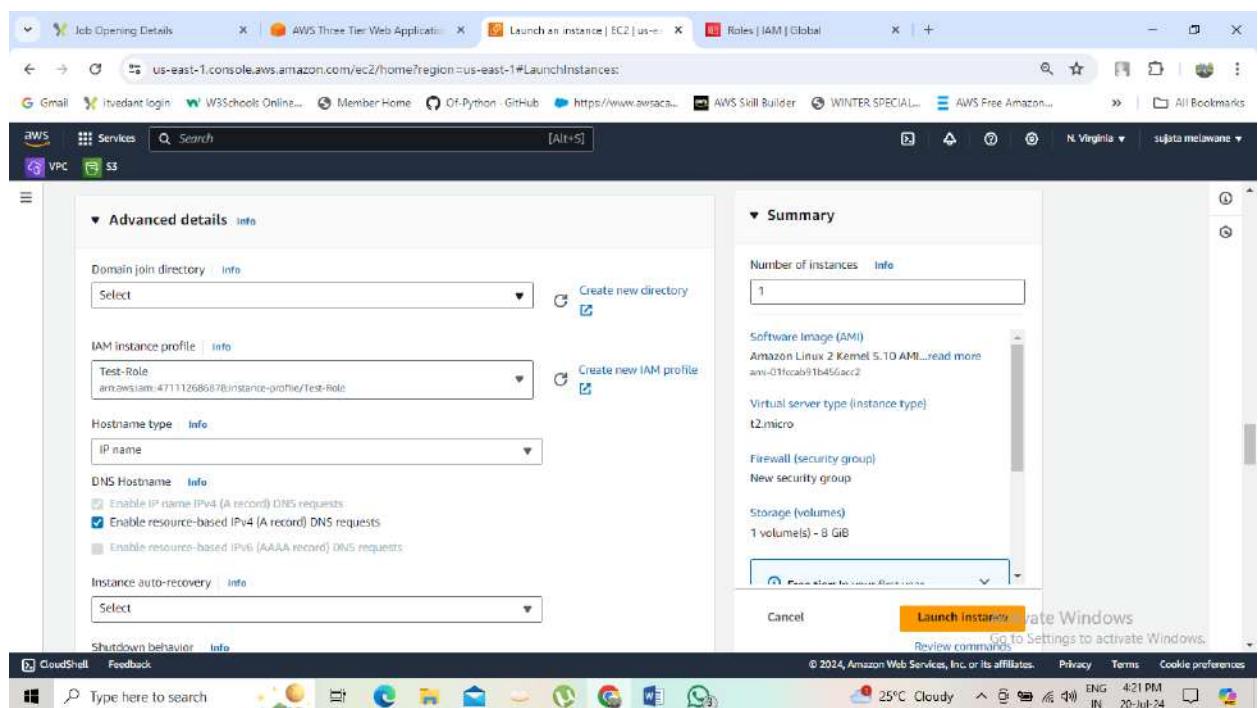
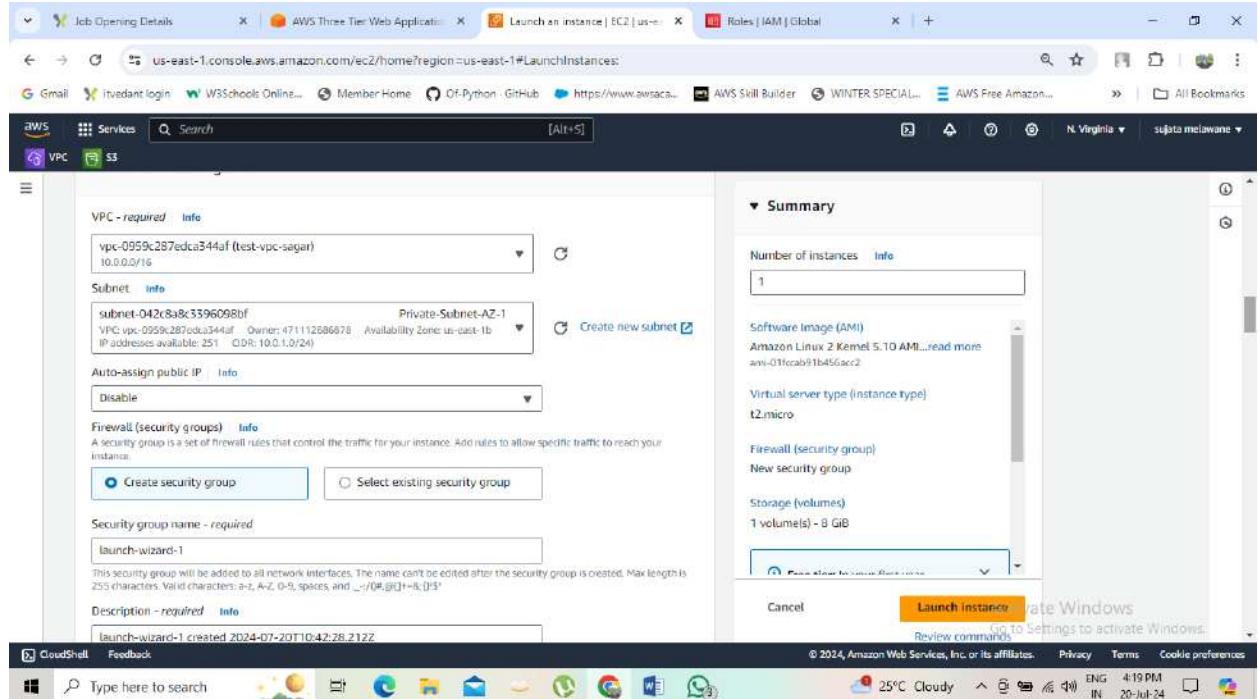
2. Select the first Amazon Linux 2 AMI



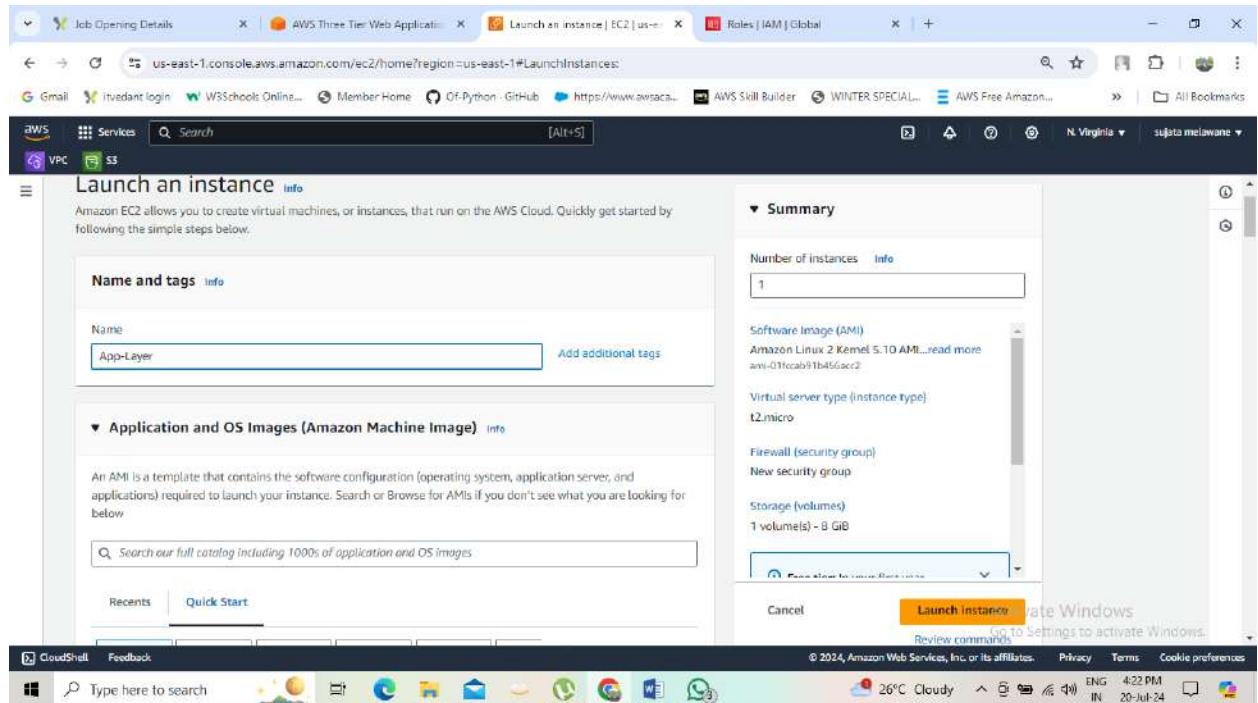
3. We'll be using the free tier eligible **T.2 micro** instance type. Select that and click **Next: Configure Instance Details**.



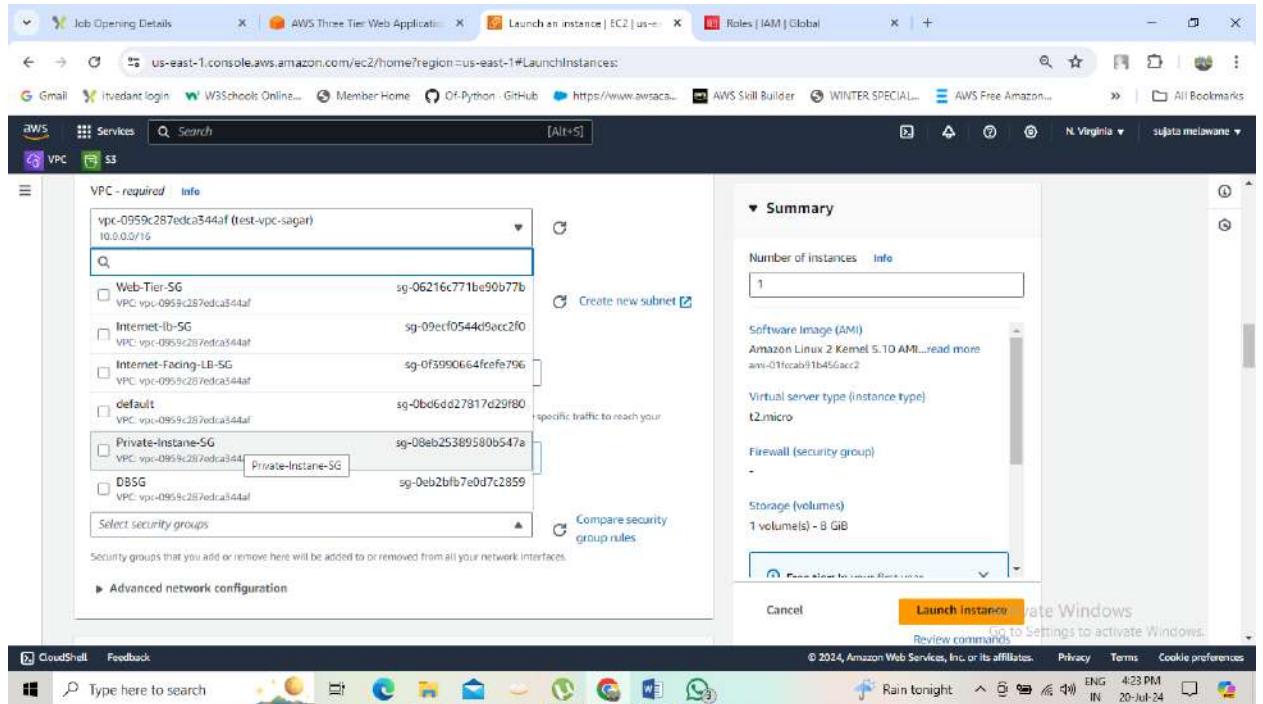
4. When configuring the instance details, make sure to select to correct **Network**, **subnet**, and **IAM role** we created. Note that this is the app layer, so use one of the private subnets we created for this layer.



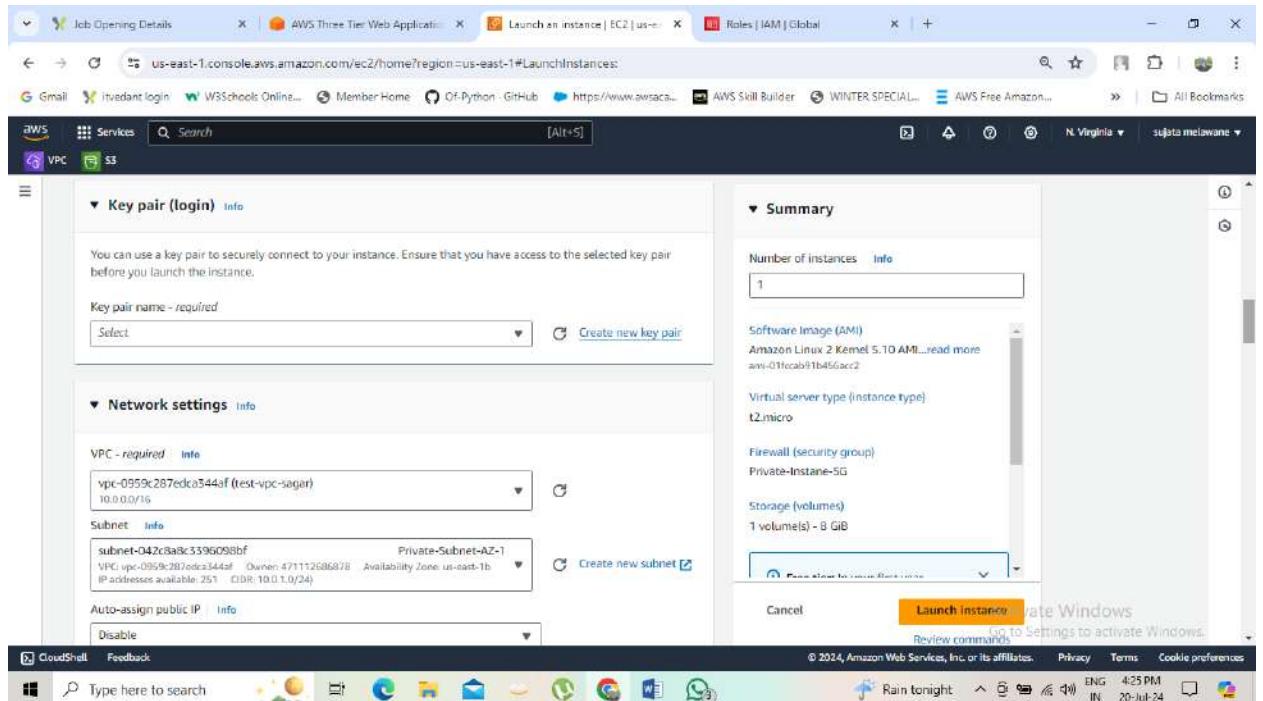
5. We'll be keeping the defaults for storage so click next twice. When you get to the tag screen input a **Name** as a key and call the instance AppLayer. It's a good idea to tag your instances so you can easily keep track of what each instance was created for. Click **Next: Configure Security Group**.



6. Earlier we created a security group for our private app layer instances, so go ahead and select that in this next section. Then click **Review and Launch**. Ignore the warning about connecting to port 22- we don't need to do that.



When you get to the **Review Instance Launch** page, review the details you configured and click **Launch**. You'll see a pop up about creating a key pair. Since we are using Systems Manager Session Manager to connect to the instance, **proceed without a keypair**. Click **Launch**

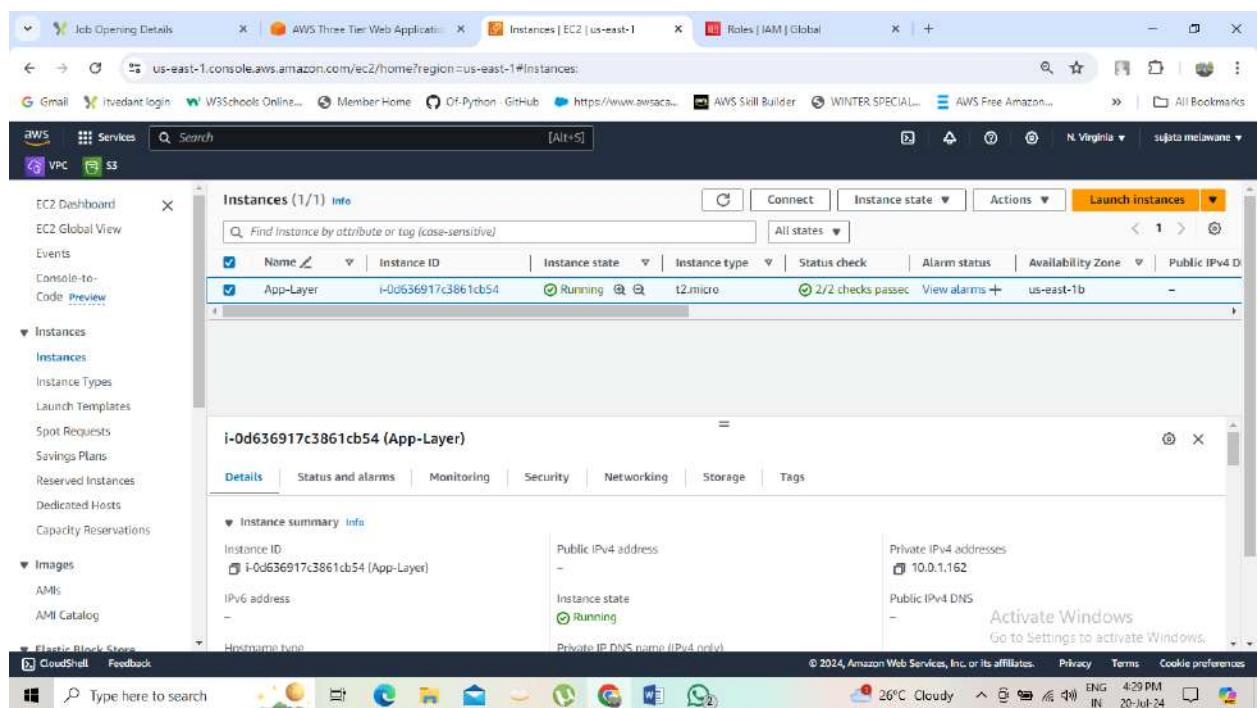


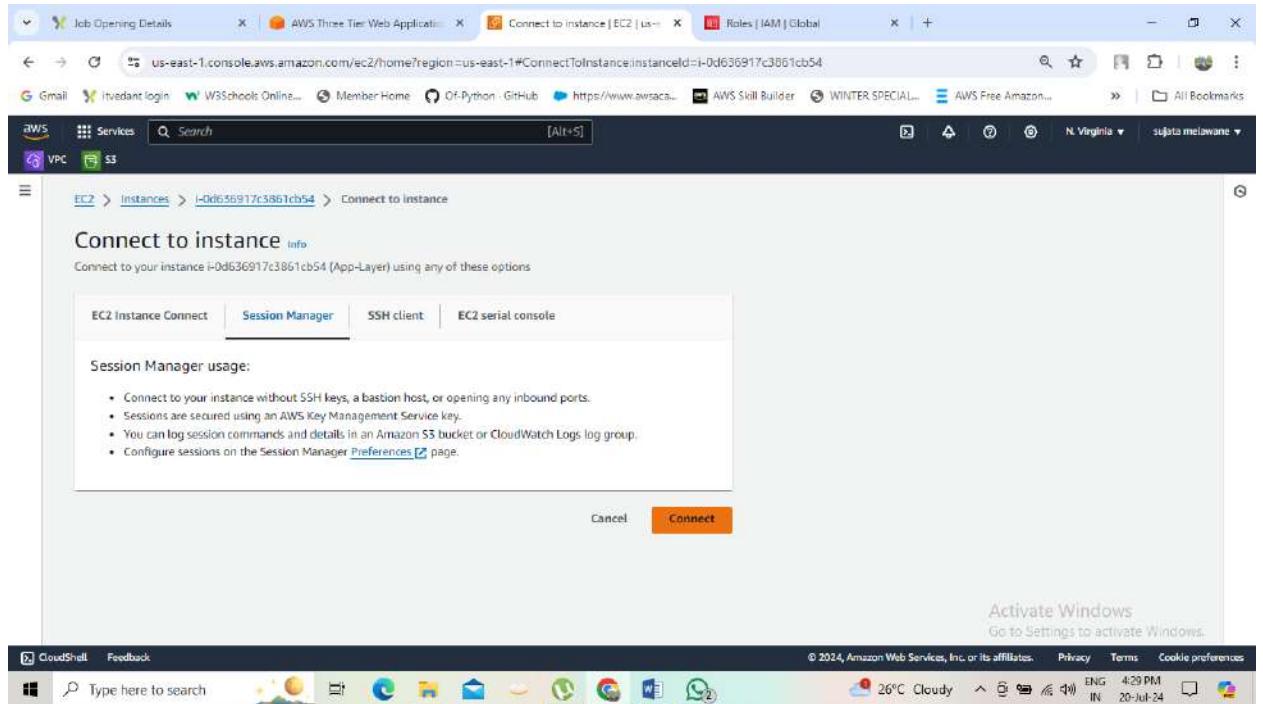
You'll be taken to a page where you can click launch instance, and you'll see the instance you just launched.

Connect to Instance

1. Navigate to your list of running EC2 Instances by clicking on **Instances** on the left hand side of the EC2 dashboard. When the instance state is running, connect to your instance by clicking the checkmark box to the left of the instance, and click the connect button on the top right corner of the dashboard. Select the Session Manager tab, and click connect. This will open a new browser tab for you.

NOTE: If you get a message saying that you cannot connect via session manager, then check that your instances can route to your NAT gateways and verify that you gave the necessary permissions on the IAM role for the Ec2 instance.

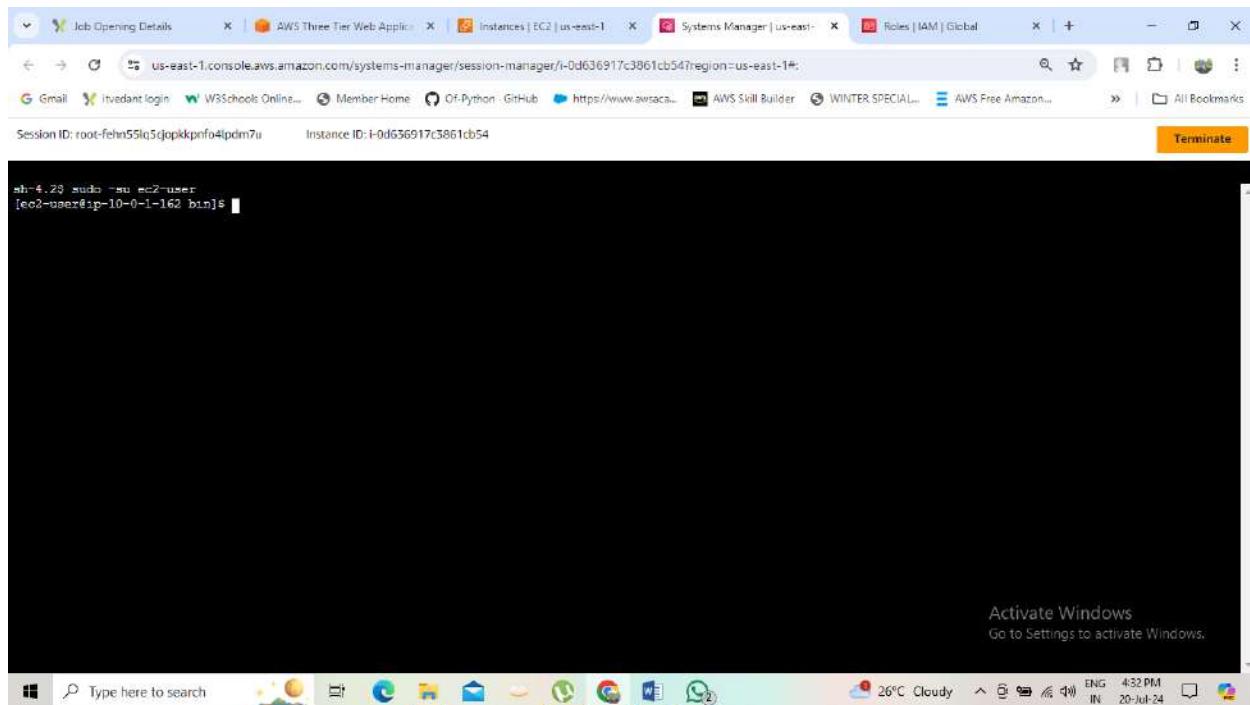




2. When you first connect to your instance like this, you will be logged in as ssm-user which is the default user. Switch to ec2-user by executing the following command in the browser terminal:

1

```
sudo -su ec2-user
```



3. Let's take this moment to make sure that we are able to reach the internet via our NAT gateways. If your network is configured correctly up till this point, you should be able to ping the google DNS servers:

1

ping 8.8.8.8

```

sh-4.2$ sudo -su ec2-user
[ec2-user@ip-10-0-1-162 bin]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=2.23 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=1.90 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=2.01 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=57 time=1.84 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=57 time=1.81 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=57 time=1.89 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=57 time=1.86 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=57 time=1.87 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=57 time=1.83 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=57 time=1.84 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=57 time=1.80 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=57 time=1.87 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=57 time=1.89 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=57 time=1.88 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=57 time=1.90 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=57 time=1.87 ms

```

Activate Windows
Go to Settings to activate Windows.

```

64 bytes from 8.8.8.8: icmp_seq=47 ttl=57 time=1.83 ms
64 bytes from 8.8.8.8: icmp_seq=48 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=49 ttl=57 time=1.84 ms
64 bytes from 8.8.8.8: icmp_seq=50 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=51 ttl=57 time=1.88 ms
64 bytes from 8.8.8.8: icmp_seq=52 ttl=57 time=1.86 ms
64 bytes from 8.8.8.8: icmp_seq=53 ttl=57 time=1.86 ms
64 bytes from 8.8.8.8: icmp_seq=54 ttl=57 time=1.83 ms
64 bytes from 8.8.8.8: icmp_seq=55 ttl=57 time=2.68 ms
64 bytes from 8.8.8.8: icmp_seq=56 ttl=57 time=1.82 ms
64 bytes from 8.8.8.8: icmp_seq=57 ttl=57 time=1.05 ms
64 bytes from 8.8.8.8: icmp_seq=58 ttl=57 time=1.84 ms
64 bytes from 8.8.8.8: icmp_seq=59 ttl=57 time=1.84 ms
64 bytes from 8.8.8.8: icmp_seq=60 ttl=57 time=1.83 ms
64 bytes from 8.8.8.8: icmp_seq=61 ttl=57 time=2.01 ms
64 bytes from 8.8.8.8: icmp_seq=62 ttl=57 time=1.95 ms
64 bytes from 8.8.8.8: icmp_seq=63 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=64 ttl=57 time=1.89 ms
64 bytes from 8.8.8.8: icmp_seq=65 ttl=57 time=1.86 ms
64 bytes from 8.8.8.8: icmp_seq=66 ttl=57 time=1.86 ms
64 bytes from 8.8.8.8: icmp_seq=67 ttl=57 time=1.84 ms
64 bytes from 8.8.8.8: icmp_seq=68 ttl=57 time=1.04 ms
64 bytes from 8.8.8.8: icmp_seq=69 ttl=57 time=1.00 ms
64 bytes from 8.8.8.8: icmp_seq=70 ttl=57 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=71 ttl=57 time=1.87 ms
64 bytes from 8.8.8.8: icmp_seq=72 ttl=57 time=1.83 ms
64 bytes from 8.8.8.8: icmp_seq=73 ttl=57 time=1.89 ms
64 bytes from 8.8.8.8: icmp_seq=74 ttl=57 time=1.83 ms
*c
--- 8.8.8.8 ping statistics ---
74 packets transmitted, 74 received, 0% packet loss, time 73133ms
rtt min/avg/max/mdev = 1.802/1.891/2.811/0.159 ms
[ec2-user@ip-10-0-1-162 bin]$ 
```

Activate Windows
Go to Settings to activate Windows.

You should see a transmission of packets. Stop it by pressing cntrl c.

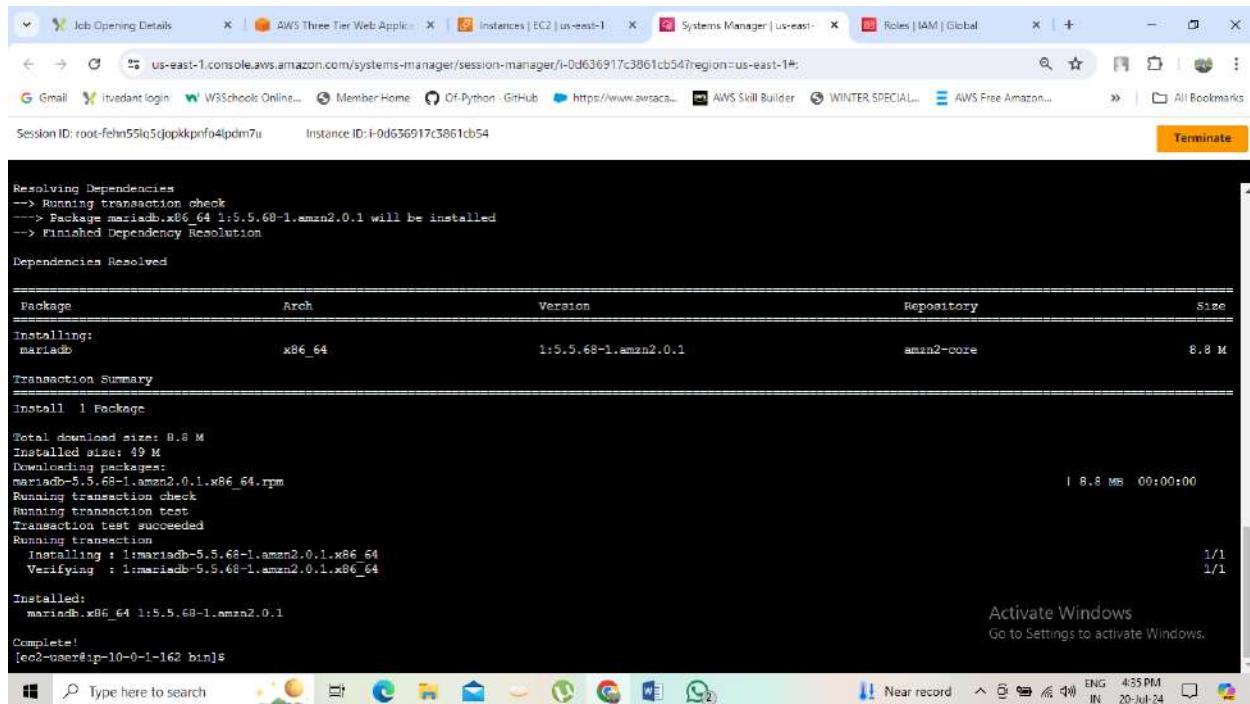
NOTE: If you can't reach the internet then you need to double check your route tables and subnet associations to verify if traffic is being routed to your NAT gateway!

Configure Database

1. Start by downloading the MySQL CLI:

```
1
```

```
sudo yum install mysql -y
```



```
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Transaction Summary
Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm | 8.8 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1
  Verifying : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-10-0-1-162 bin]$
```

2. Initiate your DB connection with your Aurora RDS writer endpoint. In the following command, replace the RDS writer endpoint and the username, and then execute it in the browser terminal:

```
1
```

```
mysql -h CHANGE-TO-YOUR-RDS-ENDPOINT -u CHANGE-TO-USER-NAME -p
```

note:- mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p

pass-Bhokare11

You will then be prompted to type in your password. Once you input the password and hit enter, you should now be connected to your database.

NOTE: If you cannot reach your database, check your credentials and security groups.

The screenshot shows a terminal window on an AWS CloudWatch interface. The session ID is root-fehn55iq5jopikkpinfo4lpdm7u and the instance ID is i-0d636917c3861cb547. The terminal output shows the following steps:

```
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package           Arch      Version            Repository      Size
Installing:
mariadb          x86_64   1:5.5.68-1.amzn2.0.1    amzn2-core      8.8 M

Transaction Summary
Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm
Running transaction check
Running transaction test
transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64      1/1
  Verifying  : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64      1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-10-0-1-162 ~]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-USER-NAME -p
Enter password: [REDACTED]
```

The terminal prompt shows the user is connected to the database:

```
[ec2-user@ip-10-0-1-162 ~]$
```

You will then be prompted to type in your password. Once you input the password and hit enter, you should now be connected to your database.

```
mysql -h CHANGE-TO-YOUR-RDS-ENDPOINT -u CHANGE-TO-USER-NAME -p
```

note:- mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p

pass-Bhokare11

```

Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-database-1 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-database-1 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-ENDPOINT -u CHANGE-TO-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-ENDPOINT' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-i-0d636917c3861cb54 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 469
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> 
```

You will then be prompted to type in your password. Once you input the password and hit enter, you should now be connected to your database.

NOTE: If you cannot reach your database, check your credentials and security groups

3. Create a database called **webappdb** with the following command using the MySQL CLI:

1

CREATE DATABASE webappdb;

```

Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-database-1-instance-1
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-database-1-instance-1 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-ENDPOINT -u CHANGE-TO-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-ENDPOINT' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-App-Layer -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-ENDPOINT -u CHANGE-TO-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-ENDPOINT' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-1-0d636917c3861cb54 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \q.
Your MySQL connection id is 469
Server version: 8.0.32 Source distribution

copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE DATABASE weebpdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> 
```

You can verify that it was created correctly with the following command:

```

1
SHOW DATABASES;

Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-ENDPOINT -u CHANGE-TO-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-ENDPOINT' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-1-0d636917c3861cb54 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 bin]$ mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \q.
Your MySQL connection id is 469
Server version: 8.0.32 Source distribution

copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE DATABASE weebpdb;
Query OK, 1 row affected (0.00 sec)

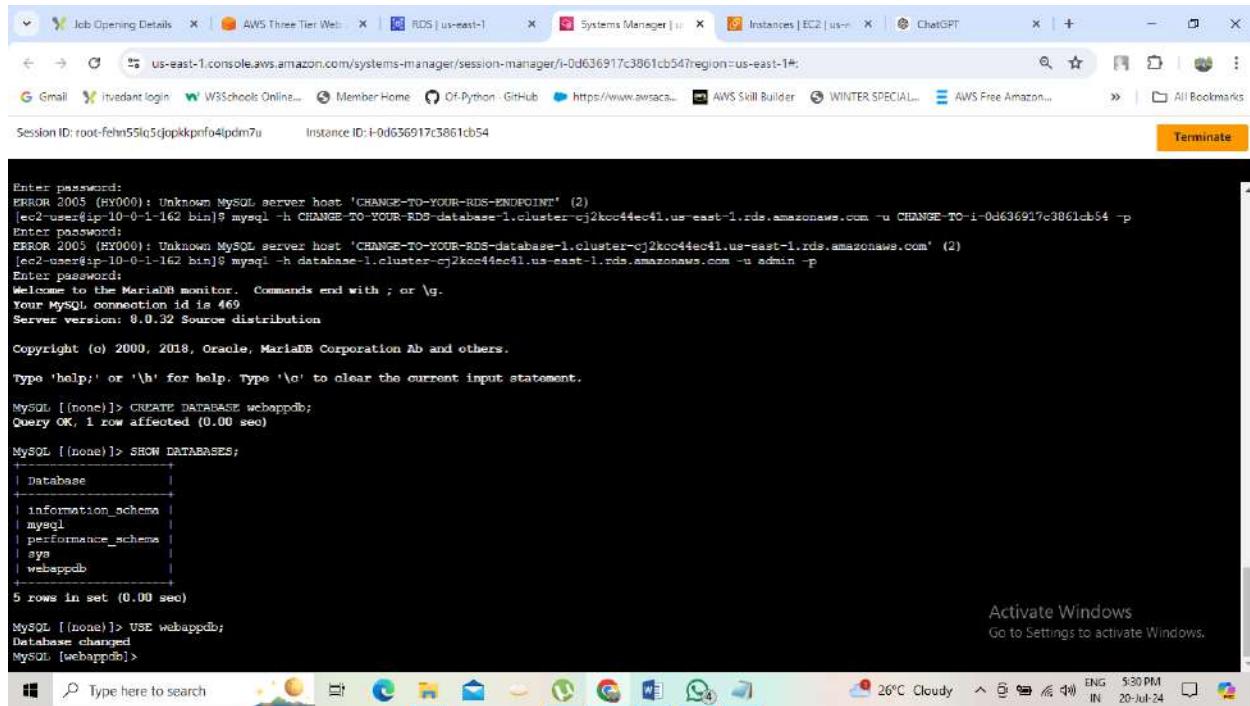
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| weebpdb |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> 
```

4. Create a data table by first navigating to the database we just created:

1

USE webappdb;



The screenshot shows a terminal window within the AWS Systems Manager interface. The user has run the 'SHOW DATABASES' command, which lists the databases: information_schema, mysql, performance_schema, sys, and webappdb. The 'webappdb' database is selected with the 'USE webappdb;' command. The terminal also displays the MySQL prompt and various error messages related to connecting to the RDS endpoint.

```
Enter password:
ERROR 2003 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-ENDPOINT' (2)
[ec2-user@ip-10-0-1-162 ~]$ mysql -h CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u CHANGE-TO-i-0d636917c3861cb54 -p
Enter password:
ERROR 2003 (HY000): Unknown MySQL server host 'CHANGE-TO-YOUR-RDS-database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com' (2)
[ec2-user@ip-10-0-1-162 ~]$ mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 469
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE DATABASE webappdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| webappdb |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]>
```

Then, create the following **transactions** table by executing this create table command:

1

2

3

```
CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL
AUTO_INCREMENT, amount DECIMAL(10,2), description
VARCHAR(100), PRIMARY KEY(id));
```

```
[ec2-user@ip-10-0-1-16Z bin]$ mysql -h database-1.cluster-cj2kcc44ec41.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 469
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE DATABASE webappdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database          |
+--------------------+
| information_schema|
| mysql              |
| performance_schema|
| sys                |
| webappdb           |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]> CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL
-> AUTO_INCREMENT, amount DECIMAL(10,2), description
-> VARCHAR(100), PRIMARY KEY(id));
Query OK, 0 rows affected (0.02 sec)

MySQL [webappdb]>
```

Verify the table was created:

1
SHOW TABLES;

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE DATABASE webappdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database          |
+--------------------+
| information_schema|
| mysql              |
| performance_schema|
| sys                |
| webappdb           |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]> CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL
-> AUTO_INCREMENT, amount DECIMAL(10,2), description
-> VARCHAR(100), PRIMARY KEY(id));
Query OK, 0 rows affected (0.02 sec)

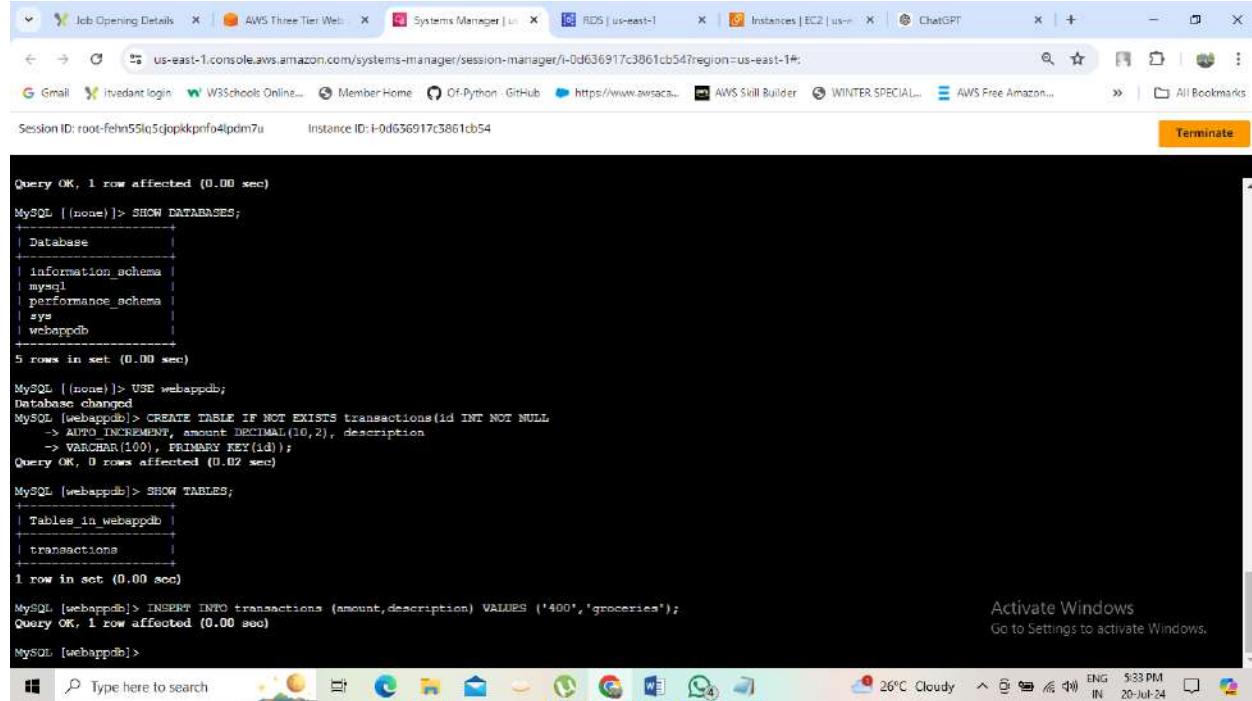
MySQL [webappdb]> SHOW TABLES;
+---------------------+
| Tables_in_webappdb |
| transactions        |
+---------------------+
1 row in set (0.00 sec)

MySQL [webappdb]>
```

5. Insert data into table for use/testing later:

1

```
INSERT INTO transactions (amount,description) VALUES ('400','groceries');
```



```
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| webappdb |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]> CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL
-> AUTO_INCREMENT, amount DECIMAL(10,2), description
-> VARCHAR(100), PRIMARY KEY(id));
Query OK, 0 rows affected (0.02 sec)

MySQL [webappdb]> SHOW TABLES;
+-----+
| Tables_in_webappdb |
+-----+
| transactions |
+-----+
1 row in set (0.00 sec)

MySQL [webappdb]> INSERT INTO transactions (amount,description) VALUES ('400','groceries');
Query OK, 1 row affected (0.00 sec)

MySQL [webappdb]>
```

Verify that your data was added by executing the following command:

1

```
SELECT * FROM transactions;
```

```
| performance_schema |
| sys |
| webappdb |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]> CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL
-> AUTO_INCREMENT, amount DECIMAL(10,2), description
-> VARCHAR(100), PRIMARY KEY(id));
Query OK, 0 rows affected (0.02 sec)

MySQL [webappdb]> SHOW TABLES;
+-----+
| Tables_in_webappdb |
+-----+
| transactions |
+-----+
1 row in set (0.00 sec)

MySQL [webappdb]> INSERT INTO transactions (amount,description) VALUES ('400','groceries');
Query OK, 1 row affected (0.00 sec)

MySQL [webappdb]> SELECT * FROM transactions;
+----+----+-----+
| id | amount | description |
+----+----+-----+
| 1 | 400.00 | groceries |
+----+----+-----+
1 row in set (0.01 sec)

MySQL [webappdb]>
```

Activate Windows
Go to Settings to activate Windows.

When finished, just type **exit** and hit enter to exit the MySQL client

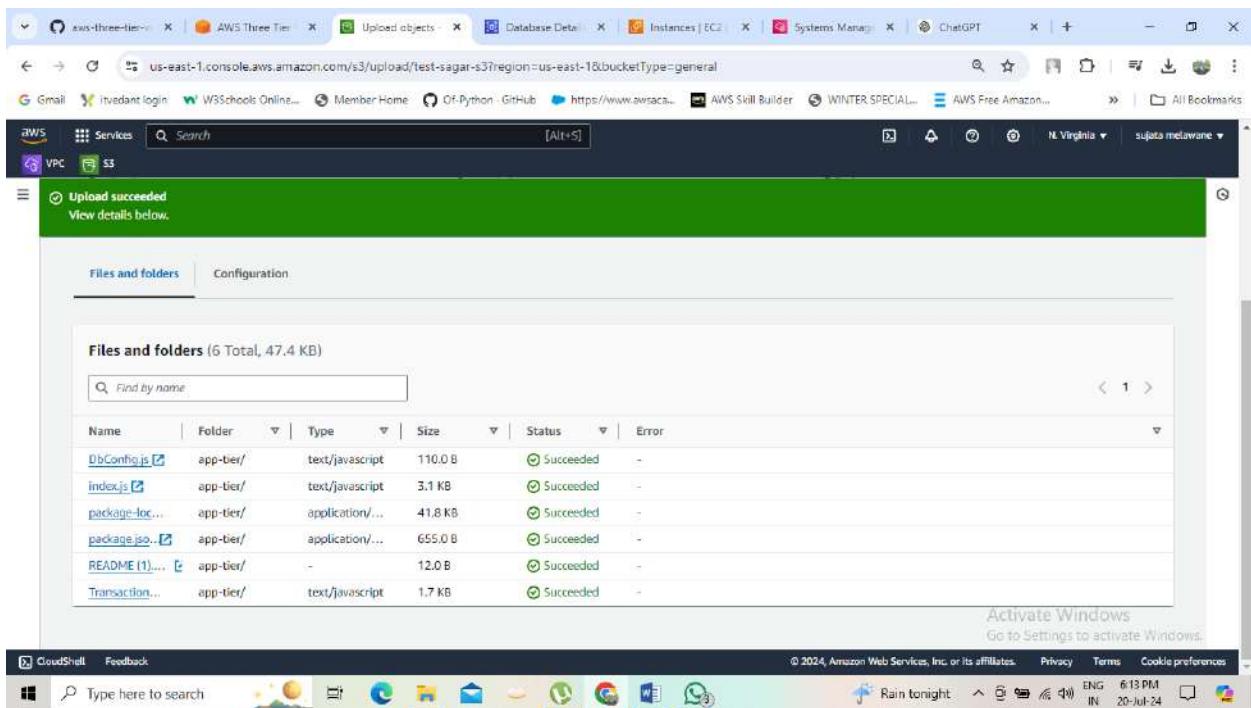
Configure App Instance

1. The first thing we will do is update our database credentials for the app tier. To do this, open the **application-code/app-tier/DbConfig.js** file from the github repo in your favorite text editor on your computer. You'll see empty strings for the hostname, user, password and database. Fill this in with the credentials you configured for your database, the **writer** endpoint of your database as the hostname, and **webappdb** for the database. Save the file.

NOTE: This is NOT considered a best practice, and is done for the simplicity of the lab. Moving these credentials to a more suitable place like Secrets Manager is left as an extension for this workshop.

The screenshot shows the AWS S3 console interface. The left sidebar is titled "Amazon S3" and includes sections for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, and Storage Lens. Under Storage Lens, there are links for Dashboards, Storage Lens groups, and AWS Organizations settings. The main content area shows the "test-sagar-s3" bucket. The navigation bar indicates the path: Amazon S3 > Buckets > test-sagar-s3. Below the navigation bar, tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points are visible. The "Objects" tab is selected, showing a table with one row: "No objects". A message below the table states, "You don't have any objects in this bucket." At the top of the objects table is a "Upload" button. The status bar at the bottom right shows "Activate Windows Go to Settings to activate Windows." The taskbar at the bottom includes icons for CloudShell, Feedback, Type here to search, and various system icons.

The screenshot shows the AWS S3 upload interface. The left sidebar is identical to the previous screenshot. The main content area has a message: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)". Below this is a large input field with the placeholder "Drag and drop files and folders you want to upload here, or choose Add files or Add folder." A "Files and folders" table follows, showing six items: DbConfig.js, index.js, package-lock.json, package.json, README (1).md, and TransactionService.js. The table includes columns for Name, Folder, and Type. The status bar at the bottom right shows "Activate Windows Go to Settings to activate Windows." The taskbar at the bottom includes icons for CloudShell, Feedback, Type here to search, and various system icons.



2. Upload the **app-tier** folder to the S3 bucket that you created in part 0.
3. Go back to your SSM session. Now we need to install all of the necessary components to run our backend application. Start by installing NVM (node version manager).
- 4.
5. `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash`
6. `source ~/.bashrc`

```

sh-4.2$ ec2-user
sh: ec2-user: command not found
sh-4.2$ sudo -su ec2-user
[ec2-user@ip-10-0-1-162 bin]$ curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
  % Total    % Received =Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent  Left Speed
100 14926  100 14926    0     0  159k  0:--:-- --:--:--:--:-- 160k
=> Downloading nvm as script to '/home/ec2-user/.nvm'

=> Appending nvm source string to /home/ec2-user/.bashrc
=> Appending bash_completion source string to /home/ec2-user/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && . "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-1-162 bin]$ source ~/.bashrc
[ec2-user@ip-10-0-1-162 bin]$

```

Activate Windows
Go to Settings to activate Windows.

4. Next, install a compatible version of Node.js and make sure it's being used

1
2

nvm install 16
nvm use 16

```

sh-4.2$ ec2-user
sh: ec2-user: command not found
sh-4.2$ sudo -su ec2-user
[ec2-user@ip-10-0-1-162 bin]$ curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
  % Total    % Received =Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent  Left Speed
100 14926  100 14926    0     0  159k  0:--:-- --:--:--:--:-- 160k
=> Downloading nvm as script to '/home/ec2-user/.nvm'

=> Appending nvm source string to /home/ec2-user/.bashrc
=> Appending bash_completion source string to /home/ec2-user/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && . "$NVM_DIR/bash_completion"
[ec2-user@ip-10-0-1-162 bin]$ source ~/.bashrc
[ec2-user@ip-10-0-1-162 bin]$ nvm install 16
Downloading and installing node v16.20.2...
Downloaded https://nodejs.org/dist/v16.20.2/node-v16.20.2-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v16.20.2 (npm v8.19.4)
Creating default alias: default -> v16.20.2
[ec2-user@ip-10-0-1-162 bin]$ nvm use 16
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-1-162 bin]$


```

Activate Windows
Go to Settings to activate Windows.

5. PM2 is a daemon process manager that will keep our node.js app running when we exit the instance or if it is rebooted. Install that as well.

1

`npm install -g pm2`

```

⇒ Appending nvm source string to /home/ec2-user/.bashrc
⇒ Appending bash_completion source string to /home/ec2-user/.bashrc
⇒ Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && . "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-1-162 bin]$ source ~/.bashrc
[ec2-user@ip-10-0-1-162 bin]$ nvm install 16
Downloading and installing node v16.20.2...
Downloading https://nodejs.org/dist/v16.20.2/node-v16.20.2-linux-x64.tar.xz...
######################################################################## 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v16.20.2 (npm v8.19.4)
Creating default alias: default -> 16 (-> v16.20.2)
[ec2-user@ip-10-0-1-162 bin]$ nvm use 16
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-1-162 bin]$ npm install -g pm2
added 138 packages, and audited 139 packages in 7s

13 packages are looking for funding
  run "npm fund" for details

found 0 vulnerabilities
npm notice New major version of npm available! 8.19.4 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice Run npm install -g npm@10.8.2 to update!
npm
[ec2-user@ip-10-0-1-162 bin]$
```

6. Now we need to download our code from our s3 buckets onto our instance. In the command below, replace BUCKET_NAME with the name of the bucket you uploaded the **app-tier** folder to:

1

2

`cd ~/`

`aws s3 cp s3://BUCKET_NAME/app-tier/ app-tier --recursive`

```

=> Close and reopen your terminal to start using nvm or run the following to use it now:
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-1-162 bin]$ source ~/.bashrc
[ec2-user@ip-10-0-1-162 bin]$ nvm install 16
Downloading and installing node v16.20.2...
Downloaded https://nodejs.org/dist/v16.20.2/node-v16.20.2-linux-x64.tar.xz...
Computing checksum with sha256sum
Checksums matched!
Now using node v16.20.2 (npm v8.19.4)
Creating default alias: default -> 16 (> v16.20.2)
[ec2-user@ip-10-0-1-162 bin]$ nvm use 16
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-1-162 bin]$ npm install -g pm2
added 138 packages, and audited 139 packages in 7s

13 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

npm notice New major version of npm available! 8.15.4 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice Run npm install -g npm@10.8.2 to update!
npm notice

[ec2-user@ip-10-0-1-162 bin]$ cd ~
[ec2-user@ip-10-0-1-162 ~]$ aws s3 cp s3://BUCKET NAME/app-tier/ app-tier --recursive
fatal error: An error occurred (AllAccessDisabled) when calling the ListObjectsV2 operation: All access to this object has been disabled
[ec2-user@ip-10-0-1-162 ~]$

Activate Windows
Go to Settings to activate Windows.

27°C Cloudy  ENG 6:20 PM
IN 20-Jul-24

```

7. Navigate to the app directory, install dependencies, and start the app with pm2.

- 1
- 2
- 3

```

cd ~/app-tier
npm install
pm2 start index.js

```

The screenshot shows a Windows desktop environment. In the center is a terminal window titled 'aws-three-tier-' with the URL 'us-east-1.console.aws.amazon.com/systems-manager/session-manager/I-0d636917c3861cb54?region=us-east-1#'. The terminal displays PM2 documentation and a command-line session where PM2 is being started and monitored.

```
Runtime Edition

PM2 is a Production Process Manager for Node.js applications
with a built-in Load Balancer.

Start and Daemonize any application:
$ pm2 start app.js

Load Balance 4 instances of api.js:
$ pm2 start api.js -i 4

Monitor in production:
$ pm2 monitor

Make pm2 auto-boot at server restart:
$ pm2 startup

To go further checkout:
http://pm2.io/
```

```
[PM2] Spawning PM2 daemon with pm2 home=/home/ec2-user/.pm2
[PM2] PM2 Successfully daemonized
[PM2] [ERROR] Script not found: /home/ec2-user/app-tier/index.js
[ec2-user@ip-10-0-1-162 app-tier]$
```

To make sure the app is running correctly run the following:

1
pm2 list

The screenshot shows a Windows desktop environment. In the center is a terminal window titled 'aws-three-tier-' with the URL 'us-east-1.console.aws.amazon.com/systems-manager/session-manager/I-0d636917c3861cb54?region=us-east-1#'. The terminal displays PM2 documentation and a command-line session where PM2 is being started and monitored. At the bottom, the 'pm2 list' command is run, displaying a table of running processes.

```
Runtime Edition

PM2 is a Production Process Manager for Node.js applications
with a built-in Load Balancer.

Start and Daemonize any application:
$ pm2 start app.js

Load Balance 4 instances of api.js:
$ pm2 start api.js -i 4

Monitor in production:
$ pm2 monitor

Make pm2 auto-boot at server restart:
$ pm2 startup

To go further checkout:
http://pm2.io/
```

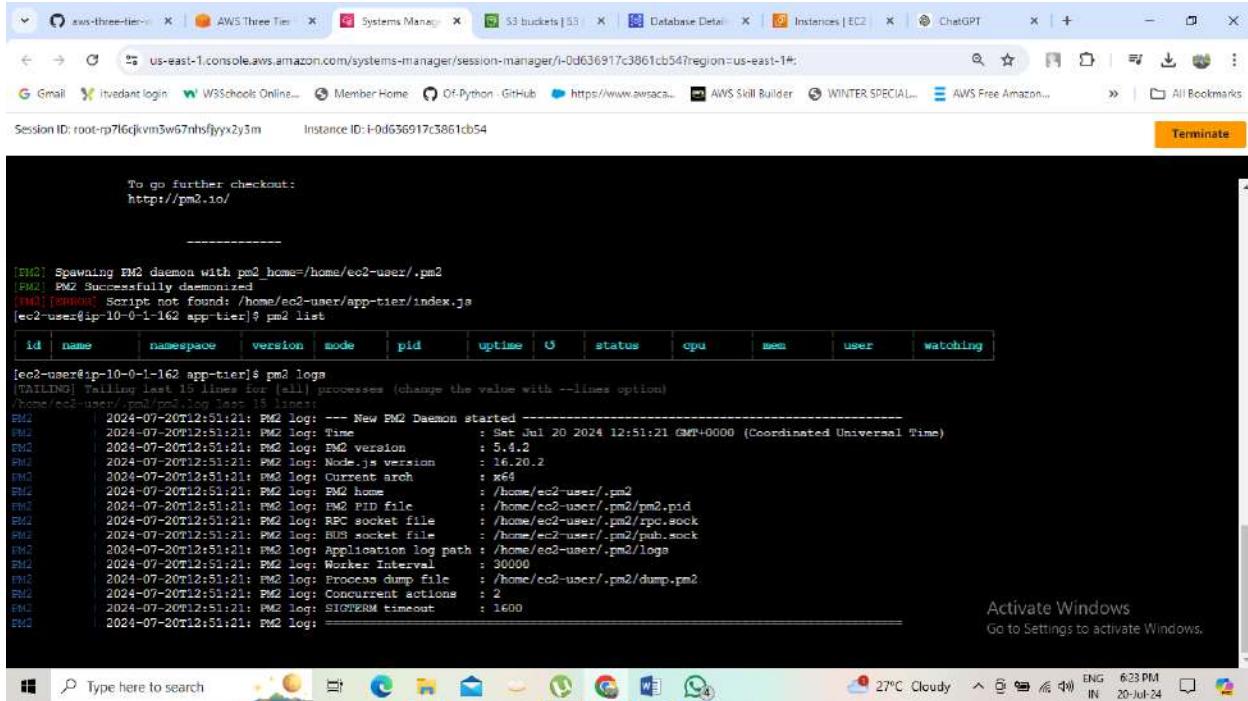
```
[PM2] Spawning PM2 daemon with pm2 home=/home/ec2-user/.pm2
[PM2] PM2 Successfully daemonized
[PM2] [ERROR] Script not found: /home/ec2-user/app-tier/index.js
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 list
```

ID	name	namespace	version	mode	pid	uptime	status	cpu	mem	user	watching

```
[ec2-user@ip-10-0-1-162 app-tier]$
```

If you see a status of online, the app is running. If you see errored, then you need to do some troubleshooting. To look at the latest errors, use this command:

1
pm2 logs



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

```
To go further checkout:  
http://pm2.io/  
  
[PM2] Spawning PM2 daemon with pm2 home=/home/ec2-user/.pm2  
[PM2] PM2 Successfully daemonized  
[PM2] Script not found: /home/ec2-user/app-tier/index.js  
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 list  
id | name | namespace | version | mode | pid | uptime | ⚡ | status | cpu | mem | user | watching  
----|----|----|----|----|----|----|----|----|----|----|----|----  
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 log  
Tailing last 16 lines for [all] processes (change the value with --lines option)  
/home/ec2-user/.pm2/logs 18 lines:  
PM2 2024-07-20T12:51:21: PM2 log: ---- New PM2 Daemon started ----  
PM2 2024-07-20T12:51:21: PM2 log: Time : Sat Jul 20 2024 12:51:21 GMT+0000 (Coordinated Universal Time)  
PM2 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2  
PM2 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2  
PM2 2024-07-20T12:51:21: PM2 log: Current arch : x64  
PM2 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2  
PM2 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pid  
PM2 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock  
PM2 2024-07-20T12:51:21: PM2 log: BUB socket file : /home/ec2-user/.pm2/pub.sock  
PM2 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs  
PM2 2024-07-20T12:51:21: PM2 log: Workers Interval : 30000  
PM2 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2  
PM2 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2  
PM2 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600  
PM2 2024-07-20T12:51:21: PM2 log: ----  
Activate Windows  
Go to Settings to activate Windows.
```

NOTE: If you're having issues, check your configuration file for any typos, and double check that you have followed all installation commands till now.

8. Right now, pm2 is just making sure our app stays running when we leave the SSM session. However, if the server is interrupted for some reason, we still want the app to start and keep running. This is also important for the AMI we will create:
9. pm2 startup

```

[PM2] [ERROR] Script not found: /home/ec2-user/app-tier/index.js
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 list


| id                                           | name | namespace | version | mode | pid | uptime | ø | status | cpu | mem | user | watching |
|----------------------------------------------|------|-----------|---------|------|-----|--------|---|--------|-----|-----|------|----------|
| [ec2-user@ip-10-0-1-162 app-tier]\$ pm2 logs |      |           |         |      |     |        |   |        |     |     |      |          |


[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/home/ec2-user/.pm2/crash.log: 15 lines:
PM2 2024-07-20T12:51:21: PM2 log: ---- New PM2 Daemon started -----
PM2 2024-07-20T12:51:21: PM2 log: Time : Set Jul 20 2024 12:51:21 GMT+0000 (Coordinated Universal Time)
PM2 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2
PM2 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pm2.pid
PM2 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 2024-07-20T12:51:21: PM2 log: EOS socket file : /home/ec2-user/.pm2/pub.sock
PM2 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 2024-07-20T12:51:21: PM2 log: ----

pm2 startup
^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user
--hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$
```

After running this you will see a message similar to this.

[PM2] To setup the Startup Script, copy/paste the following command: sudo env PATH=\$PATH:/home/ec2-user/.nvm/versions/node/v16.0.0/bin /home/ec2-user/.nvm/versions/node/v16.0.0/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user —hp /home/ec2-user

DO NOT run the above command, rather you should copy and past the command in the output you see in your own terminal. After you run it, save the current list of node processes with the following command:

1
pm2 save

```

[ec2-user@ip-10-0-1-162 app-tier]$ pm2 logs
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/home/ec2-user/.pm2/pm2.log last 15 lines:
PM2 : 2024-07-20T12:51:21: PM2 log: --- New PM2 Daemon started ---
PM2 : 2024-07-20T12:51:21: PM2 log: Time : Sat Jul 20 2024 12:51:21 GMT+0000 (Coordinated Universal Time)
PM2 : 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 : 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pm2.pid
PM2 : 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 : 2024-07-20T12:51:21: PM2 log: HUB socket file : /home/ec2-user/.pm2/pub.sock
PM2 : 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 : 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 : 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 : 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 : 2024-07-20T12:51:21: PM2 log: --

pm2 startup

^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System Found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user
--hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] [WARN] PM2 is not managing any process, skipping save...
[PM2] [WARN] To force saving use: pm2 save --force
[ec2-user@ip-10-0-1-162 app-tier]$
```

Test App Tier

Now let's run a couple tests to see if our app is configured correctly and can retrieve data from the database.

To hit out health check endpoint, copy this command into your SSM terminal. This is our simple health check endpoint that tells us if the app is simply running.

```
1
curl http://localhost:4000/health
```

```
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/home/ec2-user/.pm2/pm2.log last 15 lines:
PM2 : 2024-07-20T12:51:21: PM2 log: --- New PM2 Daemon started ---
PM2 : 2024-07-20T12:51:21: PM2 log: Time : Sat Jul 20 2024 12:51:21 GMT+0000 (Coordinated Universal Time)
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2
PM2 : 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 : 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pid
PM2 : 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 : 2024-07-20T12:51:21: PM2 log: BUSI socket file : /home/ec2-user/.pm2/pub.sock
PM2 : 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 : 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 : 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 : 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 : 2024-07-20T12:51:21: PM2 log: --

pm2 startup

^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user
--hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] [WARN] PM2 is not managing any process, skipping save...
[PM2] [WARN] To force saving use: pm2 save --force
[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/health
curl: (7) Failed to connect to localhost port 4000 after 0 ms: Couldn't connect to server
[ec2-user@ip-10-0-1-162 app-tier]$ 

Activate Windows
Go to Settings to activate Windows.
26°C Cloudy ENG 6:27 PM IN 20-Jul-24
```

```
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-ec2-user--hp...
Created symlink from /etc/systemd/system/multi-user.target.wants/pm2-ec2-user--hp.service to /etc/systemd/system/pm2-ec2-user--hp.service.
[PM2] [v] Command successfully executed.

[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unsetup systemd
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] [WARN] PM2 is not managing any process, skipping save...
[PM2] [WARN] To force saving use: pm2 save --force
[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/health
curl: (7) Failed to connect to localhost port 4000 after 0 ms: Couldn't connect to server
[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/health
curl: (7) Failed to connect to localhost port 4000 after 0 ms: Couldn't connect to server
[ec2-user@ip-10-0-1-162 app-tier]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=2.38 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=1.90 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=1.89 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=57 time=1.88 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=57 time=1.93 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=57 time=2.05 ms
```
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5006ms
rtt min/avg/max/mdev = 1.884/2.010/2.389/0.162 ms
[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/health
curl: (7) Failed to connect to localhost port 4000 after 0 ms: Couldn't connect to server
[ec2-user@ip-10-0-1-162 app-tier]$

Activate Windows
Go to Settings to activate Windows.
Construction on NH3... ENG 6:34 PM IN 20-Jul-24
```

```

100 14926 100 14926 0 0 671k 0 ---:---:---:---:--- 694k
⇒ nvm is already installed in /home/ec2-user/.nvm, trying to update the script

⇒ nvm source string already in /home/ec2-user/.bashrc
⇒ bash completion source string already in /home/ec2-user/.bashrc
⇒ Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[-s "$NVM_DIR/nvm.sh"] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[-s "$NVM_DIR/bash_completion"] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-1-162 app-tier]$ source ~/.bashrc
[ec2-user@ip-10-0-1-162 app-tier]$ nvm install 16
v16.20.2 is already installed.
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-1-162 app-tier]$ nvm use 16
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-1-162 app-tier]$ npm install -g pm2
changed 138 packages, and audited 139 packages in 6s

13 packages are looking for funding
 run 'npm fund' for details

found 0 vulnerabilities
[ec2-user@ip-10-0-1-162 app-tier]$ cd ~
[ec2-user@ip-10-0-1-162 ~]$ aws s3 cp s3://test-sagar-s3/app-tier/ app-tier --recursive
download: s3://test-sagar-s3/app-tier/DbConfig.js to app-tier/DbConfig.js
download: s3://test-sagar-s3/app-tier/README (1).md to app-tier/README (1).md
download: s3://test-sagar-s3/app-tier/TransactionService.js to app-tier/TransactionService.js
download: s3://test-sagar-s3/app-tier/package.json to app-tier/package.json
download: s3://test-sagar-s3/app-tier/index.js to app-tier/index.js
download: s3://test-sagar-s3/app-tier/package-lock.json to app-tier/package-lock.json
[ec2-user@ip-10-0-1-162 ~]$ Activate Windows
Go to Settings to activate Windows.

```

cd ~  
aws s3 cp s3://test-sagar-s3/app-tier/ app-tier –recursive

```

found 0 vulnerabilities
[ec2-user@ip-10-0-1-162 app-tier]$ cd ~
[ec2-user@ip-10-0-1-162 ~]$ aws s3 cp s3://test-sagar-s3/app-tier/ app-tier --recursive
download: s3://test-sagar-s3/app-tier/DbConfig.js to app-tier/DbConfig.js
download: s3://test-sagar-s3/app-tier/README (1).md to app-tier/README (1).md
download: s3://test-sagar-s3/app-tier/TransactionService.js to app-tier/TransactionService.js
download: s3://test-sagar-s3/app-tier/package.json to app-tier/package.json
download: s3://test-sagar-s3/app-tier/index.js to app-tier/index.js
download: s3://test-sagar-s3/app-tier/package-lock.json to app-tier/package-lock.json
[ec2-user@ip-10-0-1-162 ~]$ cd ~/app-tier
[ec2-user@ip-10-0-1-162 app-tier]$ npm install
added 69 packages, and audited 69 packages in 2s

2 packages are looking for funding
 run 'npm fund' for details

3 high severity vulnerabilities

To address all issues, run:
 npm audit fix

Run 'npm audit' for details.
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 start index.js
[PM2] Starting /home/ec2-user/app-tier/index.js in fork_mode (1 instance)
[PM2] Done.

[PM2] [WARN] Current process list is not synchronized with saved list. Type 'pm2 save' to synchronize.
[ec2-user@ip-10-0-1-162 app-tier]$ Activate Windows
Go to Settings to activate Windows.

```

Session ID: root-rp7l6ckvm3w67nhsfjyyx5m    Instance ID: i-0d636917c3861cb54

**Terminate**

```

download: s3://test-sagar-s3/app-tier/index.js to app-tier/index.js
download: s3://test-sagar-s3/app-tier/package-lock.json to app-tier/package-lock.json
[ec2-user@ip-10-0-1-162 ~]$ cd ~/app-tier
[ec2-user@ip-10-0-1-162 app-tier]$ npm install

added 60 packages, and audited 60 packages in 2s

2 packages are looking for funding
 run 'npm fund' for details

3 high severity vulnerabilities

To address all issues, run:
 npm audit fix

Run 'npm audit' for details.
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 start index.js
[PM2] Starting /home/ec2-user/app-tier/index.js in fork_mode (1 instance)
[PM2] Done.

ID	Name	Namespace	Version	Mode	pid	Uptime	Status	CPU	Mem	User	Watching	
0	index	default	1.0.0	[●]	2227	0s	0	online	0%	23.0mb	ec2-user	disabled

[PM2] [WARN] Current process list is not synchronized with saved list. Type 'pm2 save' to synchronize.
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 list

ID	Name	Namespace	Version	Mode	pid	Uptime	Status	CPU	Mem	User	Watching	
0	index	default	1.0.0	[●]	2227	29s	0	online	0%	52.2mb	ec2-user	disabled


```

Activate Windows  
Go to Settings to activate Windows.

Rainy tomorrow    ENG IN 20-Jul-24

Session ID: root-rp7l6ckvm3w67nhsfjyyx5m    Instance ID: i-0d636917c3861cb54

**Terminate**

```

[PM2] [WARN] Current process list is not synchronized with saved list. Type 'pm2 save' to synchronize.
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 list

ID	Name	Namespace	Version	Mode	pid	Uptime	Status	CPU	Mem	User	Watching	
0	index	default	1.0.0	[●]	2227	29s	0	online	0%	52.2mb	ec2-user	disabled

[PM2] [WARN] Current process list is not synchronized with saved list. Type 'pm2 save' to synchronize.
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 log
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/home/ec2-user/.pm2/logs/index.log last 15 lines:
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2
PM2 : 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 : 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pid
PM2 : 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 : 2024-07-20T12:51:21: PM2 log: BUS socket file : /home/ec2-user/.pm2/pub.sock
PM2 : 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 : 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 : 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 : 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 : =====
PM2 : 2024-07-20T13:29:43: PM2 log: App [index:0] starting in -fork mode-
PM2 : 2024-07-20T13:29:43: PM2 log: App [index:0] online

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:
/home/ec2-user/.pm2/logs/index-var.log last 15 lines:
0|index | AB3 backend app listening at http://localhost:4000

```

Activate Windows  
Go to Settings to activate Windows.

Rainy tomorrow    ENG IN 20-Jul-24

```
[PM2] [WARN] Current process list is not synchronized with saved list. Type 'pm2 save' to synchronize.
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 log
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/home/ec2-user/.pm2/log last 15 lines:
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2
PM2 : 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 : 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pm2.pid
PM2 : 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 : 2024-07-20T12:51:21: PM2 log: BU9 socket file : /home/ec2-user/.pm2/pub.sock
PM2 : 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 : 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 : 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 : 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 : 2024-07-20T12:51:21: PM2 log: -----
PM2 : 2024-07-20T13:29:43: PM2 log: App [index:0] starting in -fork mode-
PM2 : 2024-07-20T13:29:43: PM2 log: App [index:0] online

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:
/home/ec2-user/.pm2/logs/index-out.log last 15 lines:
0 index | AB3 backend app listening at http://localhost:4000

pm2 startup

^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user
--hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$
```

```
[/home/ec2-user/.pm2/log last 15 lines:
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2
PM2 : 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 : 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pm2.pid
PM2 : 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 : 2024-07-20T12:51:21: PM2 log: BU9 socket file : /home/ec2-user/.pm2/pub.sock
PM2 : 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 : 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 : 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 : 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 : 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 : 2024-07-20T13:29:43: PM2 log: App [index:0] starting in -fork mode-
PM2 : 2024-07-20T13:29:43: PM2 log: App [index:0] online

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:
/home/ec2-user/.pm2/logs/index-out.log last 15 lines:
0 index | AB3 backend app listening at http://localhost:4000

pm2 startup

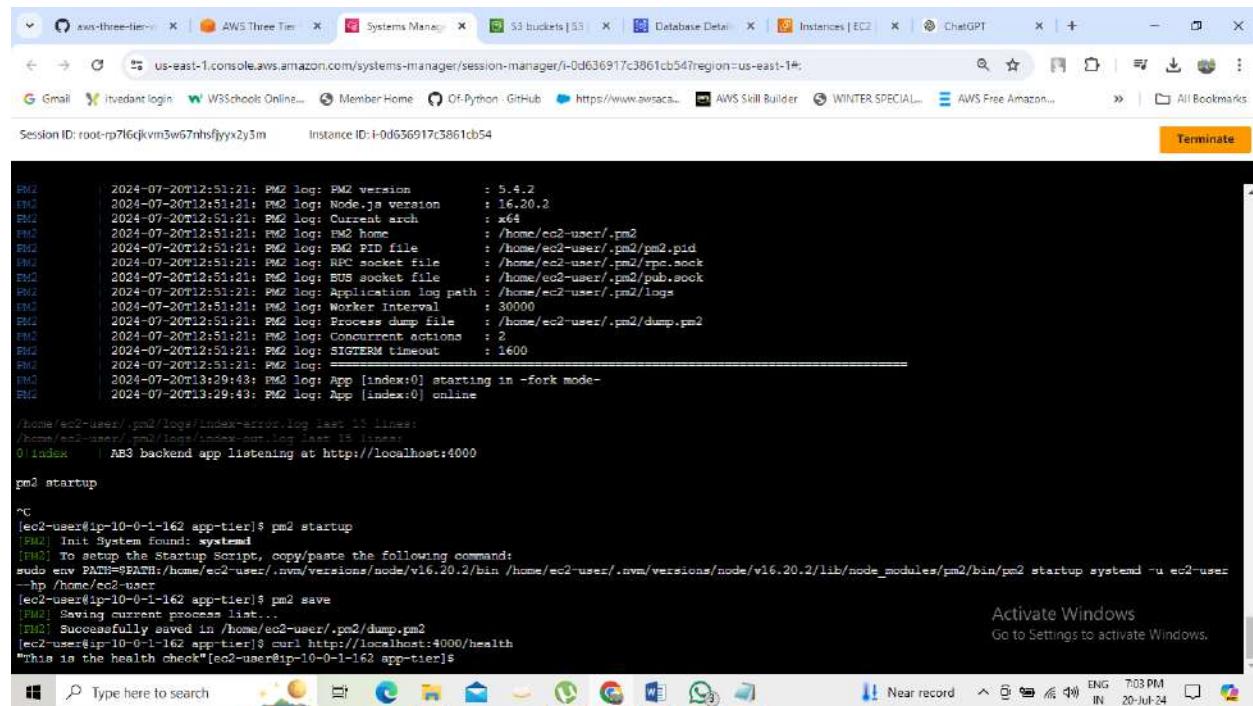
^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user
--hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[ec2-user@ip-10-0-1-162 app-tier]$
```

# Test App Tier

Now let's run a couple tests to see if our app is configured correctly and can retrieve data from the database.

To hit out health check endpoint, copy this command into your SSM terminal. This is our simple health check endpoint that tells us if the app is simply running.

```
1 curl http://localhost:4000/health
```



The screenshot shows a Windows taskbar at the bottom with various icons. Above it is a terminal window titled 'aws-three-tier' with the tab bar showing 'aws Three Tier'. The terminal content displays PM2 logs for an application, followed by a command prompt where 'pm2 startup' is run, and finally a command to save the process list. A watermark for 'Activate Windows' is visible in the bottom right corner of the terminal window.

```
PM2 2024-07-20T12:51:21: PM2 log: PM2 version : 5.4.2
PM2 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 2024-07-20T12:51:21: PM2 log: PM2 pid file : /home/ec2-user/.pm2/pm2.pid
PM2 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 2024-07-20T12:51:21: PM2 log: EOS socket file : /home/ec2-user/.pm2/pub.sock
PM2 2024-07-20T12:51:21: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
=====
PM2 2024-07-20T12:51:21: PM2 log: App [index:0] starting in -fork mode-
PM2 2024-07-20T13:29:43: PM2 log: App [index:0] online

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:
/home/ec2-user/.pm2/logs/index-out.log last 15 lines:
0 index | AB3 backend app listening at http://localhost:4000

pm2 startup

^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2_startup systemctl -u ec2-user
-hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/health
"this is the health check"[ec2-user@ip-10-0-1-162 app-tier]$
```

The response should looks like the following:

```
1
"This is the health check"
```

Next, test your database connection. You can do that by hitting the following endpoint locally:

```
1 curl http://localhost:4000/transaction
```

```

PM2 2024-07-20T12:51:21: PM2 log: Node.js version : 16.20.2
PM2 2024-07-20T12:51:21: PM2 log: Current arch : x64
PM2 2024-07-20T12:51:21: PM2 log: PM2 home : /home/ec2-user/.pm2
PM2 2024-07-20T12:51:21: PM2 log: PM2 PID file : /home/ec2-user/.pm2/pid
PM2 2024-07-20T12:51:21: PM2 log: RPC socket file : /home/ec2-user/.pm2/rpc.sock
PM2 2024-07-20T12:51:21: PM2 log: BUS socket file : /home/ec2-user/.pm2/pub.sock
PM2 2024-07-20T12:51:21: PM2 log: Application log path: /home/ec2-user/.pm2/logs
PM2 2024-07-20T12:51:21: PM2 log: Worker Interval : 30000
PM2 2024-07-20T12:51:21: PM2 log: Process dump file : /home/ec2-user/.pm2/dump.pm2
PM2 2024-07-20T12:51:21: PM2 log: Concurrent actions : 2
PM2 2024-07-20T12:51:21: PM2 log: SIGTERM timeout : 1600
PM2 2024-07-20T12:51:21: PM2 log:
PM2 2024-07-20T13:29:43: PM2 log: App [index:0] starting in -fork mode-
PM2 2024-07-20T13:29:43: PM2 log: App [index:0] online

```

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:  
/home/ec2-user/.pm2/logs/index-out.log last 10 lines:  
0|index | AB3 backend app listening at http://localhost:4000

```

pm2 startup
^C
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user
--hp /home/ec2-user
[ec2-user@ip-10-0-1-162 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/health
"this is the health check"[ec2-user@ip-10-0-1-162 app-tier]$ curl http://localhost:4000/transaction
"result": [{"id":1,"amount":400,"description":"groceries"}, {"id":2,"amount":100,"description":"class"}, {"id":3,"amount":200,"description":"other groceries"}, {"id":4,"amount":10,"description":"brownies"}]

```

Activate Windows  
Go to Settings to activate Windows.

25°C Cloudy ENG 7:04 PM IN 20-Jul-24

You should see a response containing the test data we added earlier:

```

1
{
 "result": [
 {"id":1,"amount":400,"description":"groceries"}, {"id":2,"amount":100,"description":"class"}, {"id":3,"amount":200,"description":"other groceries"}, {"id":4,"amount":10,"description":"brownies"}
]
}

```

If you see both of these responses, then your networking, security, database and app configurations are correct.

7. Congrats! Your app layer is fully configured and ready to go.

# Part 4: Internal Load Balancing and Auto Scaling

In this section of the workshop we will create an Amazon Machine Image (AMI) of the app tier instance we just created, and use that to set up autoscaling with a load balancer in order to make this tier highly available.

## Learning Objectives:

- Create an AMI of our App Tier
- Create a Launch Template
- Configure Autoscaling
- Deploy Internal Load Balancer

## App Tier AMI

1. Navigate to **Instances** on the left hand side of the EC2 dashboard. Select the app tier instance we created and under **Actions** select **Image and templates**. Click **Create Image**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation pane with options like EC2 Dashboard, Instances, Images, and AMIs. The main area displays a table of instances with one row selected: 'App-Layer' (Instance ID: i-0d636917c3861cb54). The instance is listed as 'Running' with an 't2.micro' instance type and '2/2 checks passed'. A context menu is open over this instance, with 'Create image' highlighted. Below the table, there's a detailed view for the selected instance.

- Give the image a name and description and then click **Create image**. This will take a few minutes, but if you want to monitor the status of image creation you can see it by clicking **AMIs** under **Images** on the left hand navigation panel of the EC2 dashboard.

The screenshot shows the 'Create image' wizard. Step 1: Set instance details. It asks for the 'Instance ID' (selected: i-0d636917c3861cb54), 'Image name' (entered: App Tier image), and 'Image description - optional' (entered: App Tier). There are also checkboxes for 'No reboot' and 'Enable'. The right side of the screen shows a progress bar indicating the process is 100% complete, and a message: 'Activate Windows Go to Settings to activate Windows.'

Screenshot of the AWS Cloud Console showing the creation of an Amazon Machine Image (AMI) from an EC2 instance.

The top navigation bar shows tabs for "Create Image", "Systems Manager", "S3 buckets", "Database Details", and "ChatGPT".

The main content area is titled "Instance Volumes" under the "EC2" service. It displays a table for creating new volumes:

| Storage type | Device    | Snapshot            | Size | Volume type               | IOPS | Throughput | Delete on termination               | Encrypted                |
|--------------|-----------|---------------------|------|---------------------------|------|------------|-------------------------------------|--------------------------|
| EBS          | /dev/sda1 | Create new snapshot | 8    | EBS General Purpose (SSD) | 100  | 1000       | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Below the table, a note states: "During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes."

Under "Tags - optional", there are two radio button options:

- Tag image and snapshots together: Tag the image and the snapshots with the same tag.
- Tag image and snapshots separately: Tag the image and the snapshots with different tags.

No tags are associated with the resource. An "Add new tag" button is available.

A message at the bottom left says: "You can add up to 50 more tags."

The status bar at the bottom right shows "Activate Windows" and "Go to Settings to activate Windows".

The bottom navigation bar includes links for "CloudShell", "Feedback", and search.

**Instances | EC2** tab is selected in the second screenshot.

The main content area shows a green banner: "Currently creating AMI ami-0a8bc2f80beefc06 from Instance i-0d636917c3861cb54. Check that the AMI status is Available before deleting the instance or carrying out other actions related to this AMI."

The "Instances (1/1) Info" table lists one instance:

| Name      | Instance ID         | Instance state | Instance type | Status check      | Alarm status | Availability Zone | Public IP |
|-----------|---------------------|----------------|---------------|-------------------|--------------|-------------------|-----------|
| App-Layer | i-0d636917c3861cb54 | Running        | t2.micro      | 2/2 checks passed | ...          | us-east-1b        | -         |

The instance details page for "i-0d636917c3861cb54 (App-Layer)" shows the following information:

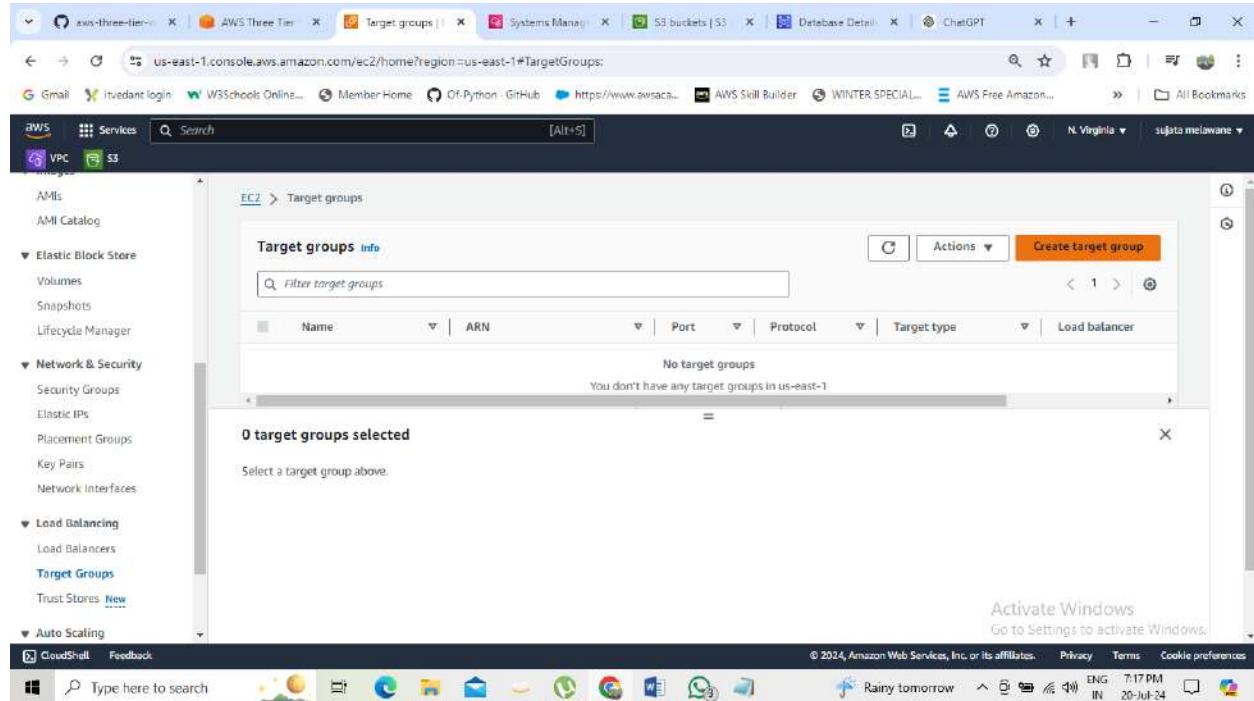
- Details** tab: Instance ID (i-0d636917c3861cb54), Public IPv4 address (-), Private IPv4 addresses (10.0.1.162), Public IPv4 DNS (-).
- Status and alarms** tab: Instance state (Running).
- Monitoring**, **Security**, **Networking**, **Storage**, and **Tags** tabs are also present.

The status bar at the bottom right shows "Activate Windows" and "Go to Settings to activate Windows".

The bottom navigation bar includes links for "CloudShell", "Feedback", and search.

# Target Group

1. While the AMI is being created, we can go ahead and create our target group to use with the load balancer. On the EC2 dashboard navigate to **Target Groups** under **Load Balancing** on the left hand side. Click on **Create Target Group**.



2. The purpose of forming this target group is to use with our load balancer so it may balance traffic across our private app tier instances. Select Instances as the target type and give it a name.

The screenshot shows the AWS Lambda function creation wizard at Step 1: Create target group. In the 'Basic configuration' section, the 'Choose a target type' step is displayed. The 'Instances' option is selected, which includes the following benefits:

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

Other options shown include:

- IP addresses**: Supports load balancing to VPC and on-premises resources, facilitates routing to multiple IP addresses and network interfaces on the same instance, offers flexibility with microservice-based architectures, and supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function**: Facilitates routing to a single Lambda function, accessible to Application Load Balancers only.
- Application Load Balancer**: Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC, facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

The screenshot shows the AWS Lambda function creation wizard at Step 1: Create target group. In the 'Basic configuration' section, the 'Choose a target type' step is displayed. The 'Lambda function' option is selected, which includes the following benefits:

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Other options shown include:

- Application Load Balancer**: Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC, facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Below these sections, the 'Target group name' field is filled with 'AppTierTargetgroup'. The 'Protocol : Port' section shows 'HTTP' selected as the protocol and '80' as the port. The 'IP address type' section indicates that only targets with the indicated IP address type can be registered to this target group, with 'IPv4' selected. A note states: 'Each function has a default endpoint (functionArn/invocationUrl) that is prefixed by the prefix you enter here.'

Then, set the protocol to **HTTP** and the port to 4000. Remember that this is the port our Node.js app is running on. Select the VPC we've been using thus far, and then change the health check path to be **/health**. This is the health check endpoint of our app. Click **Next**.

The screenshot shows the AWS Lambda console interface. A new function named "HelloWorld" is being created. The "Code" tab is active, displaying the following Python code:

```
def lambda_handler(event, context):
 """
 This is the entry point for the Lambda function.
 It handles incoming events and returns a response.
 """

 # Process the event
 print("Event received: " + str(event))

 # Create a response
 response = {"statusCode": 200, "body": "Hello World!"}

 return response
```

The "Handler" dropdown is set to "lambda\_function.lambda\_handler". The "Runtime" dropdown is set to "Python 3.9". The "Memory size" input field is set to "128". The "Timeout" input field is set to "3". The "Layers" section shows a single layer named "HelloWorldLayer".

3. We are **NOT** going to register any targets for now, so just skip that step and create the target group.

Screenshot of the AWS Cloud Console showing the "Create target group" wizard Step 2: Register targets.

The "Available instances (1)" table shows one instance:

| Instance ID         | Name      | State   | Security groups     |
|---------------------|-----------|---------|---------------------|
| i-0d636917c3861cb54 | App-Layer | Running | Private-Instance-SG |

The "Ports for the selected instances" field contains "4000".

Bottom status bar: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 7:21 PM IN 20-Jul-24

Screenshot of the AWS Cloud Console showing the "Target group details" page for the "AppTierTargetgroup".

The "Details" section shows:

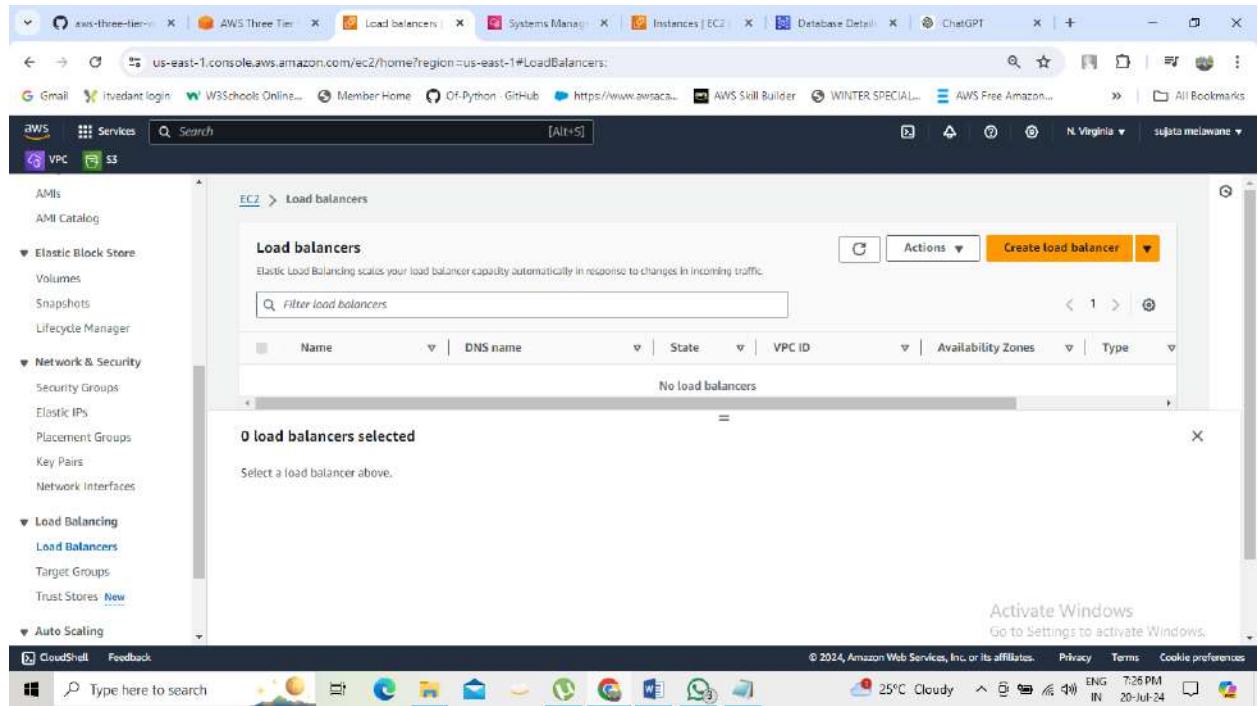
| Target type | Protocol   | Protocol version | VPC                   |
|-------------|------------|------------------|-----------------------|
| Instance    | HTTP: 4000 | HTTP1            | vpc-0959c287edca344af |

The "Targets" table shows 0 total targets, all healthy.

Bottom status bar: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 7:24 PM IN 20-Jul-24

# Internal Load Balancer

1. On the left hand side of the EC2 dashboard select **Load Balancers** under **Load Balancing** and click **Create Load Balancer**

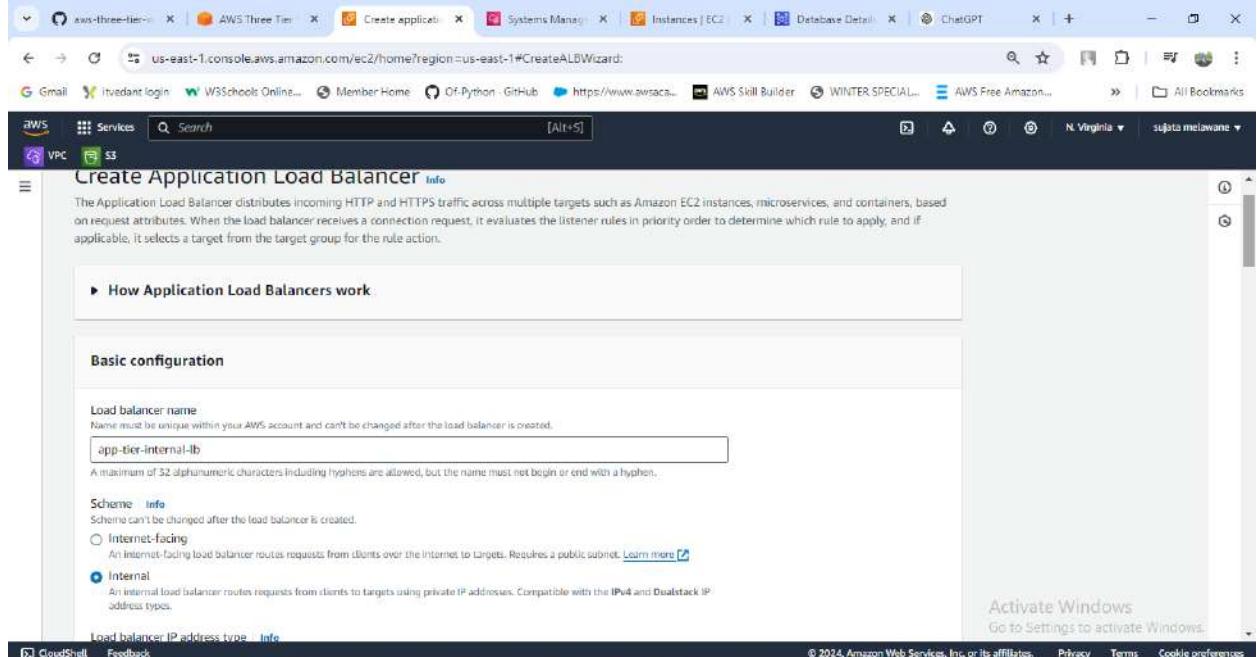


- 2.
3. We'll be using an **Application Load Balancer** for our **HTTP** traffic so click the create button for that option.

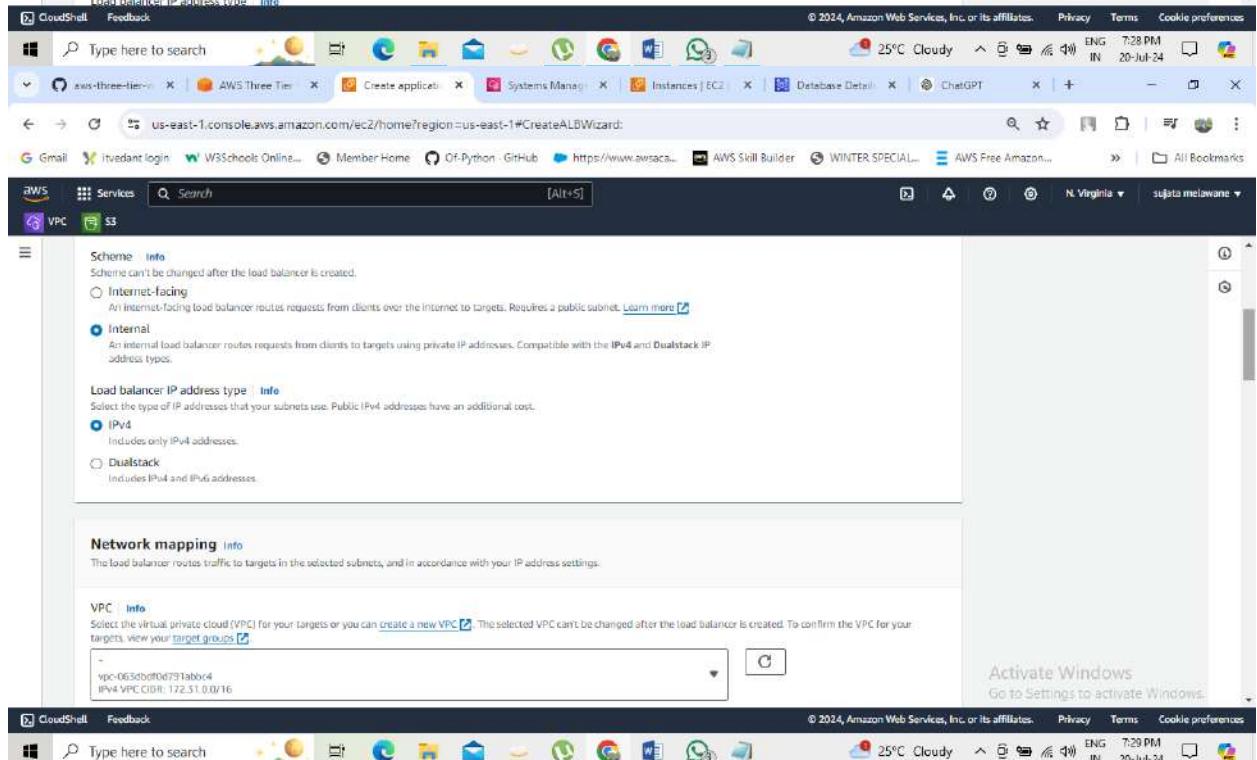


- 4.

5. After giving the load balancer a name, be sure to select **internal** since this one will not be public facing, but rather it will route traffic from our web tier to the app tier.



6.



7.

8. Select the correct network configuration for VPC and private subnets.

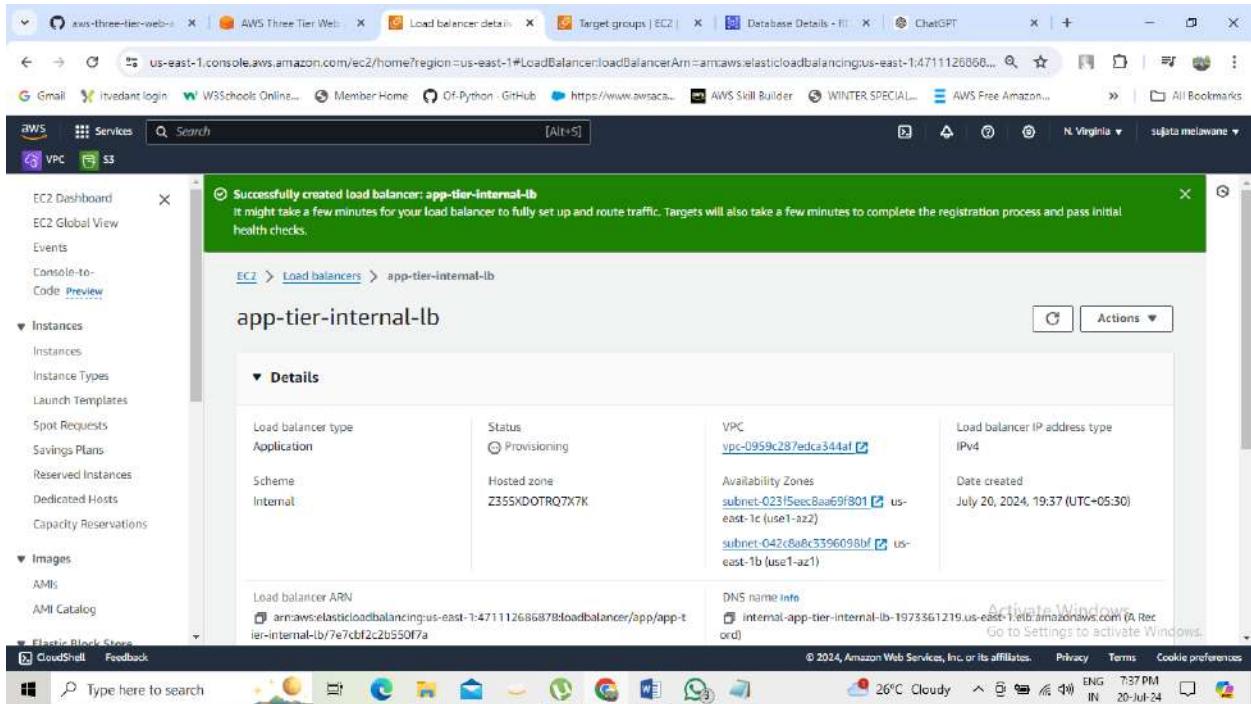
The screenshot shows the 'Network mapping' section of the AWS VPC configuration. It lists two subnets under 'Mappings': 'us-east-1b (use1-az1)' and 'us-east-1c (use1-az2)'. Both subnets are associated with the VPC 'test-vpc-s4gar' (VPC ID: vpc-0995c287edca344af, CIDR: 10.0.0.0/16). The 'us-east-1b' subnet is Private-Subnet-AZ-1 and has an IPv4 address assigned from CIDR 10.0.1.0/24. The 'us-east-1c' subnet is Private-Subnet-AZ-2 and has an IPv4 address assigned from CIDR 10.0.4.0/24.

9.

10. Select the security group we created for this internal ALB. Now, this ALB will be listening for HTTP traffic on port 80. It will be forwarding the traffic to our **target group** that we just created, so select it from the dropdown, and create the load balancer

The screenshot shows the 'Target groups' section of the AWS configuration. A new target group named 'Internet-lb-SG' is selected. Under 'Listeners and routing', there is one listener named 'Listener HTTP:80' configured to forward traffic from port 80 to an AppTier target group (HTTP, Target type: Instance, IPv4). The target group is associated with the VPC 'test-vpc-s4gar' (VPC ID: vpc-0995c287edca344af).

11.

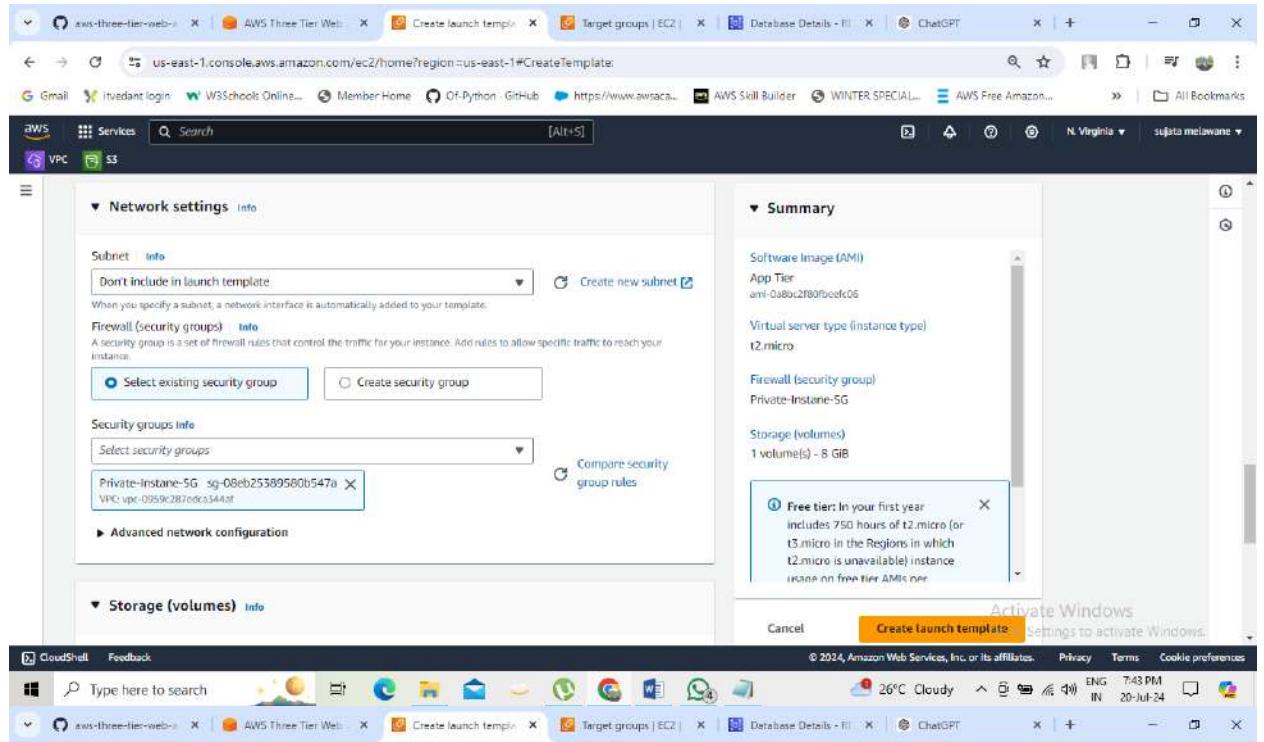


# Launch Template

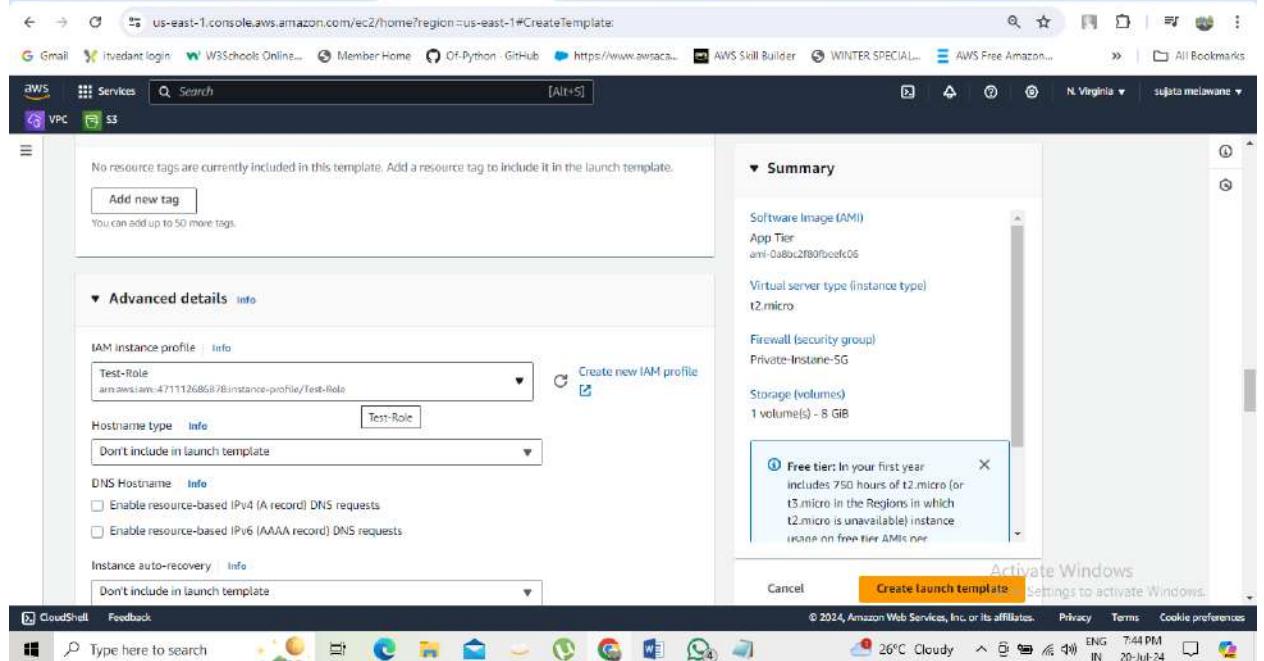
1. Before we configure Auto Scaling, we need to create a Launch template with the AMI we created earlier. On the left side of the EC2 dashboard navigate to **Launch Template** under **Instances** and click **Create Launch Template**.

2. Name the Launch Template, and then under **Application and OS Images** include the app tier AMI you created.

- 3.
4. Set the correct security group for our app tier, and then under **Advanced details** use the same IAM instance profile we have been using for our EC2 instances.



5.



6.

7.

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Create launch template" and displays a green success message: "Successfully created IAM-TEST-ROLE [lt-085e191eb8f35a424].". Below this, there's a "Next Steps" section with links to "Launch an instance", "Create an Auto Scaling group from your template", and "Create a Spot Fleet". The browser's address bar shows the URL: "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate". The AWS navigation bar at the top includes "Services", "Search", and "EC2 > Launch templates > Create launch template". The status bar at the bottom indicates "2024, Amazon Web Services, Inc. or its affiliates." and the date "20-Jul-24".

The screenshot shows a browser window displaying the "Launch Templates" list. The left sidebar shows navigation options like "EC2 Dashboard", "Instances", "Launch Templates", and "Images". The main area shows a table with one entry:

| Launch Template ID   | Launch Template Name | Default Version | Latest Version | Create Time              | Created...    |
|----------------------|----------------------|-----------------|----------------|--------------------------|---------------|
| lt-085e191eb8f35a424 | IAM-TEST-ROLE        | 1               | 1              | 2024-07-20T14:16:59.000Z | arn:aws:ia... |

A modal window is open for the "IAM-TEST-ROLE (lt-085e191eb8f35a424)". It shows the "Launch template details" section with fields: "Launch template ID" (lt-085e191eb8f35a424), "Launch template name" (IAM-TEST-ROLE), "Default version" (1), and "Owner" (arn:aws:iam::471112686878:root). The modal has "Actions" and "Delete template" buttons. The browser's address bar shows the URL: "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates". The AWS navigation bar and status bar are visible at the top and bottom respectively.

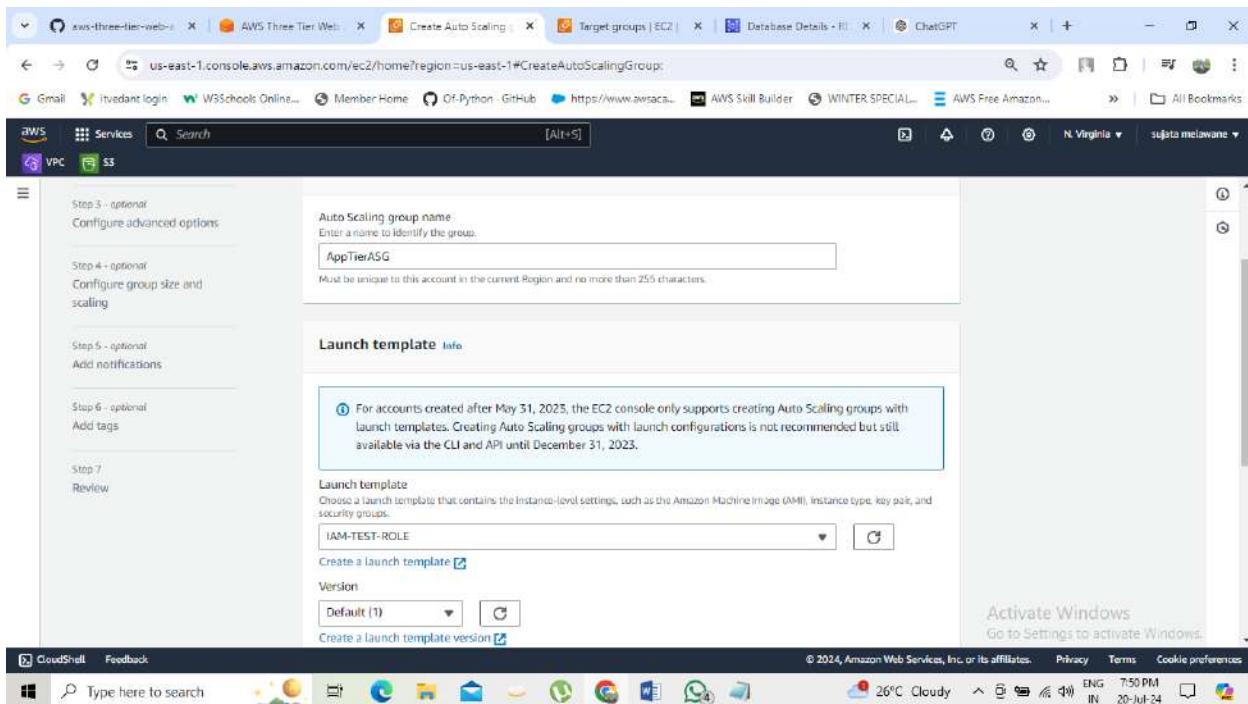
# Auto Scaling

1. We will now create the Auto Scaling Group for our app instances. On the left side of the EC2 dashboard navigate to **Auto Scaling Groups** under **Auto Scaling** and click **Create Auto Scaling group**.

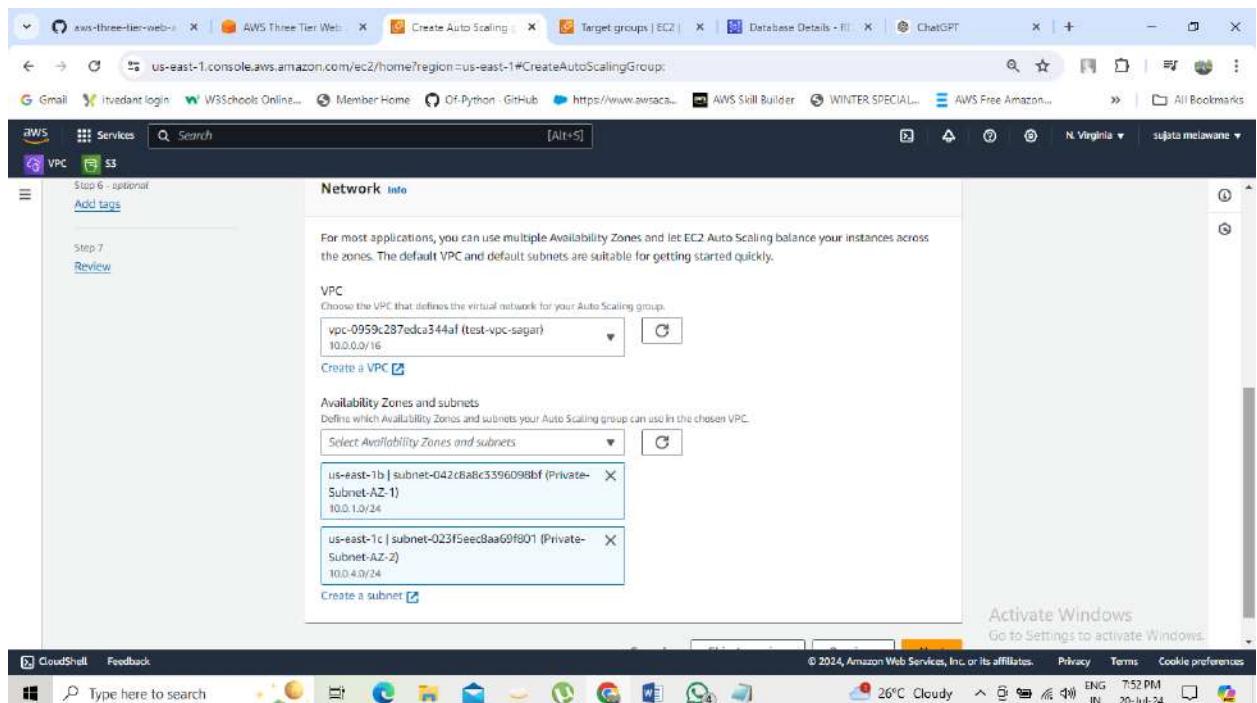
The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar navigation includes 'Elastic Block Store', 'Network & Security', 'Load Balancing', and 'Auto Scaling' (with 'Auto Scaling Groups' selected). The main content area features a large banner with the text 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. Below the banner, there's a section titled 'How it works' with a diagram showing an 'Auto Scaling group' connected to multiple EC2 instances. To the right, there's a 'Pricing' section and a call-to-action button labeled 'Create Auto Scaling group'. The browser's address bar shows the URL 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups'.

2.

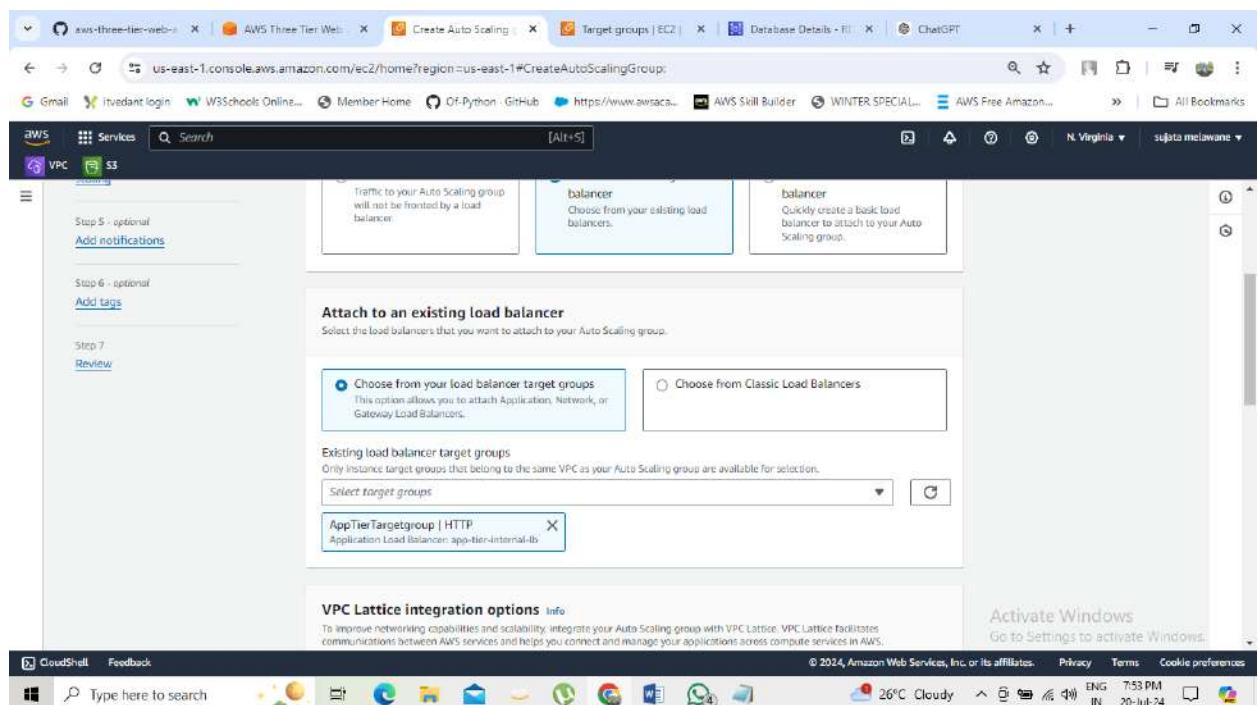
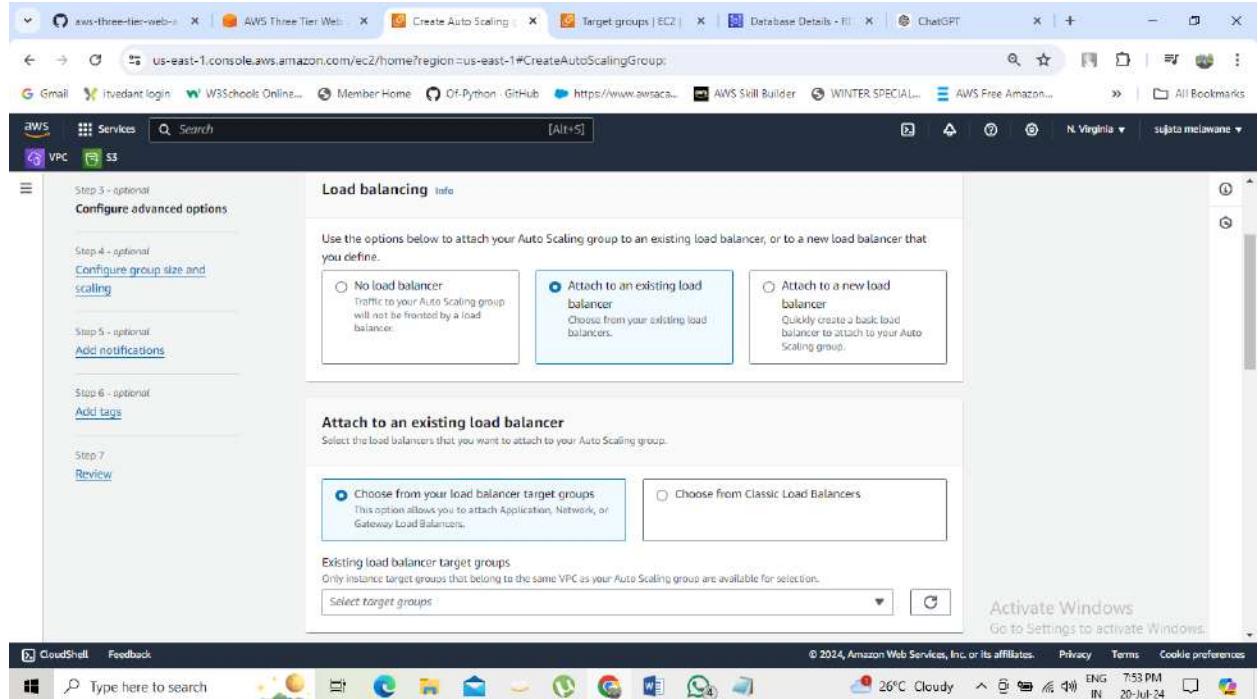
Give your Auto Scaling group a name, and then select the Launch Template we just created and click next



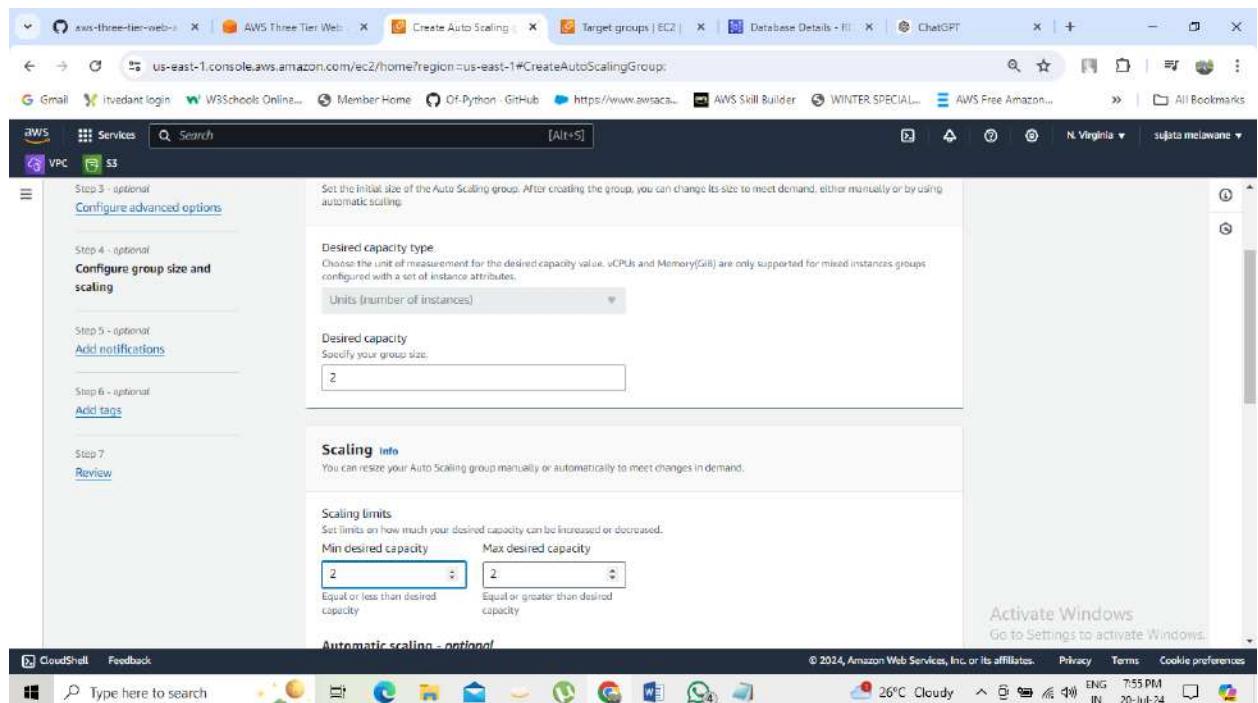
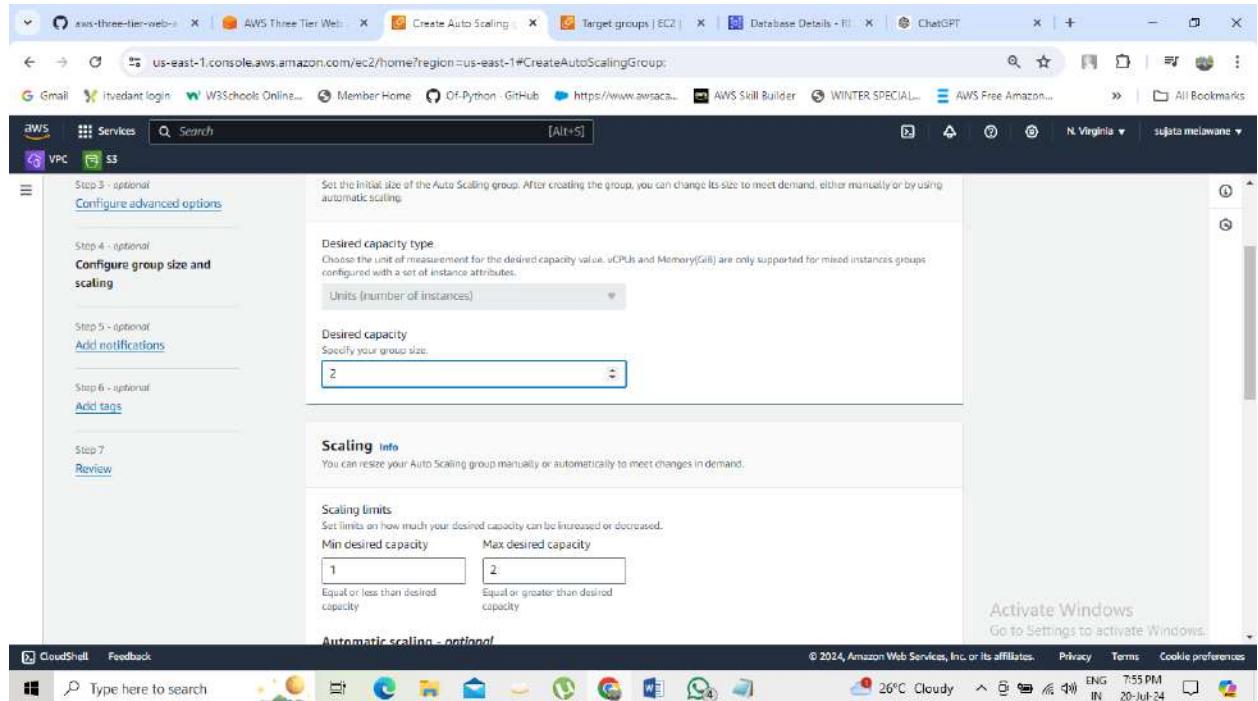
3. On the **Choose instance launch options** page set your VPC, and the private instance subnets for the app tier and continue to step 3.

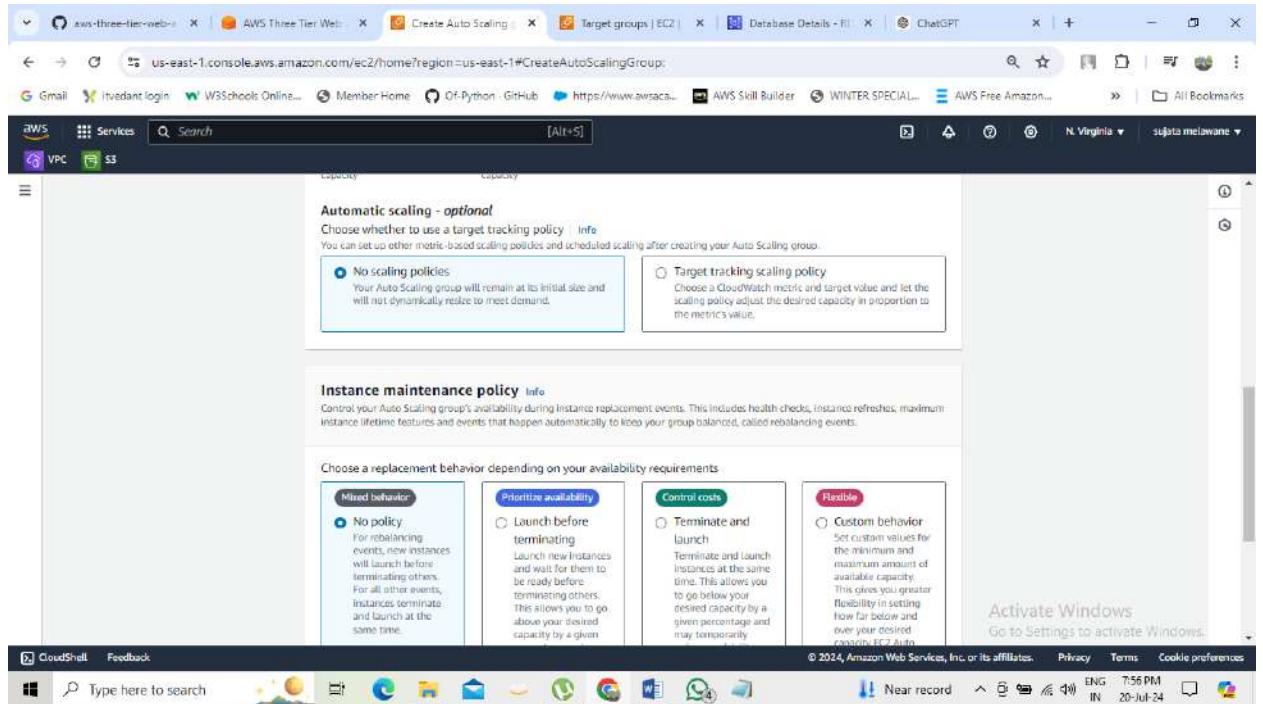


4. For this next step, attach this Auto Scaling Group to the Load Balancer we just created by selecting the existing load balancer's target group from the dropdown. Then, click next.



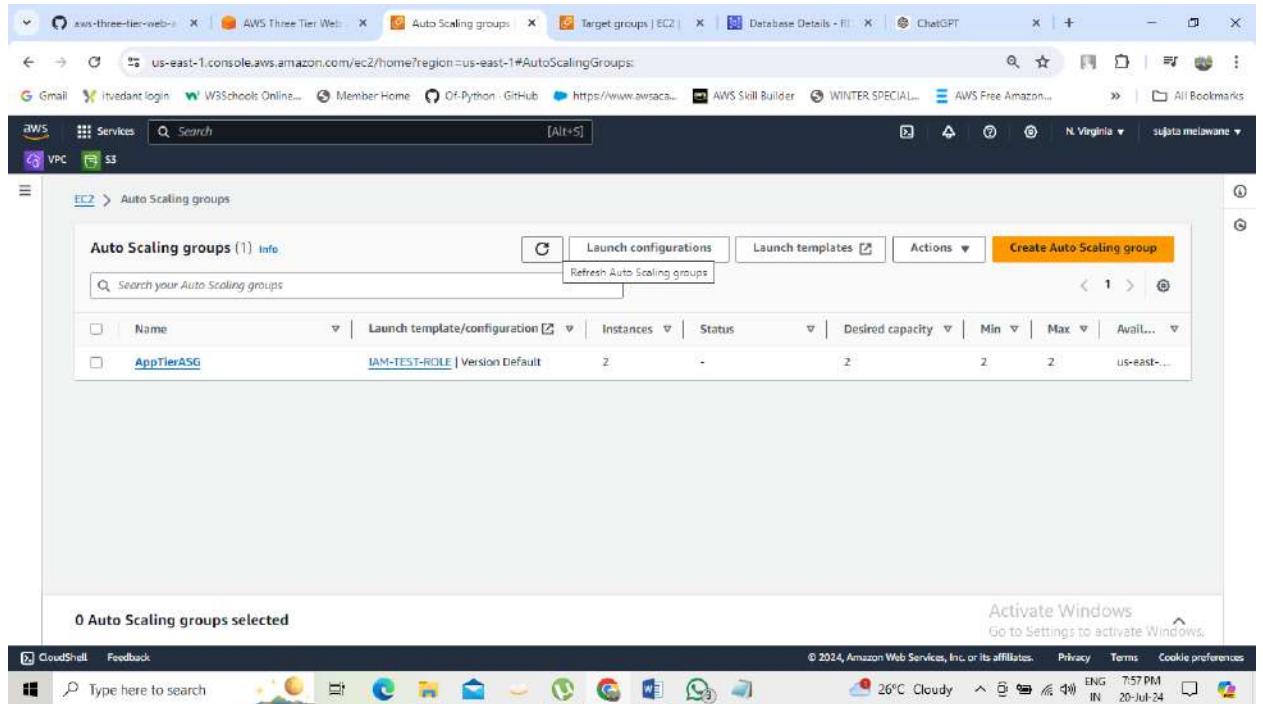
5. For **Configure group size and scaling policies**, set desired, minimum and maximum capacity to **2**. Click skip to review and then Create Auto Scaling Group.





You should now have your internal load balancer and autoscaling group configured correctly. You should see the autoscaling group spinning up 2 new app tier instances. If you wanted to test if this is working correctly, you can delete one of your new instances manually and wait to see if a new instance is booted up to replace it.

*NOTE: Your original app tier instance is excluded from the ASG so you will see 3 instances in the EC2 dashboard. You can delete your original instance that you used to generate the app tier AMI but it's recommended to keep it around for troubleshooting purposes.*



## Part 5: Web Tier Instance Deployment

In this section we will deploy an EC2 instance for the web tier and make all necessary software configurations for the NGINX web server and React.js website.

### Learning Objectives

- Update NGINX Configuration Files
- Create Web Tier Instance
- Configure Software Stack

## Update Config File

Before we create and configure the web instances, open up the **application-code/nginx.conf** file from the repo we downloaded. Scroll down to **line 58** and replace [INTERNAL-LOADBALANCER-DNS] with your internal load balancer's DNS entry. You can find this by navigating to your internal load balancer's details page.

The screenshot shows a Windows desktop environment. In the center is a code editor window titled "nginx.conf" with the following configuration:

```

17 http {
18 server {
19 location / {
20 root /home/ec2-user/web-tier/build;
21 index index.html index.htm;
22 try_files $uri /index.html;
23 }
24 #proxy for internal lb
25 location /api/ {
26 proxy_pass http://internal-app-tier-internal-lb-1973361219.us-east-1.elb.amazonaws.com:80/;
27 }
28 }
29 }

```

Below the code editor is a terminal window titled "CloudShell" showing the output of a command:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG
Loading configuration...
Done loading configuration

```

The AWS CloudShell interface is visible on the left, showing various services like VPC, S3, and Auto Scaling.

Then, upload this file **and** the **application-code/web-tier** folder to the s3 bucket you created for this lab.

The screenshot shows the AWS S3 console with a green success message: "Upload succeeded".

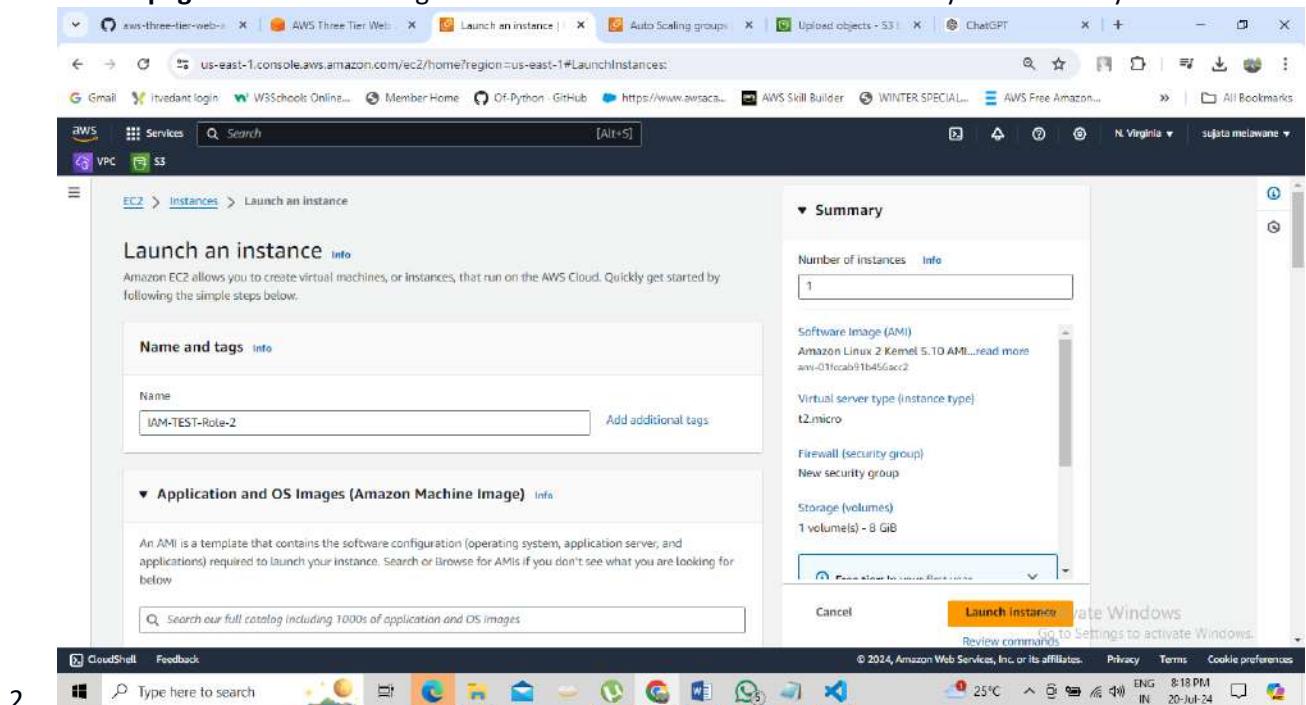
**Files and folders (7 Total, 49.9 KB)**

| Name              | Folder    | Type            | Size    | Status    | Error |
|-------------------|-----------|-----------------|---------|-----------|-------|
| DbConfig.js       | app-tier/ | text/javascript | 191.0 B | Succeeded | -     |
| index.js          | app-tier/ | text/javascript | 3.1 KB  | Succeeded | -     |
| nginx.conf        | app-tier/ | -               | 2.5 KB  | Succeeded | -     |
| package-lock.json | app-tier/ | application/... | 41.8 KB | Succeeded | -     |
| package.json      | app-tier/ | application/... | 655.0 B | Succeeded | -     |
| README(1).md      | app-tier/ | -               | 12.0 B  | Succeeded | -     |
| Transaction.js    | app-tier/ | text/javascript | 1.7 KB  | Succeeded | -     |

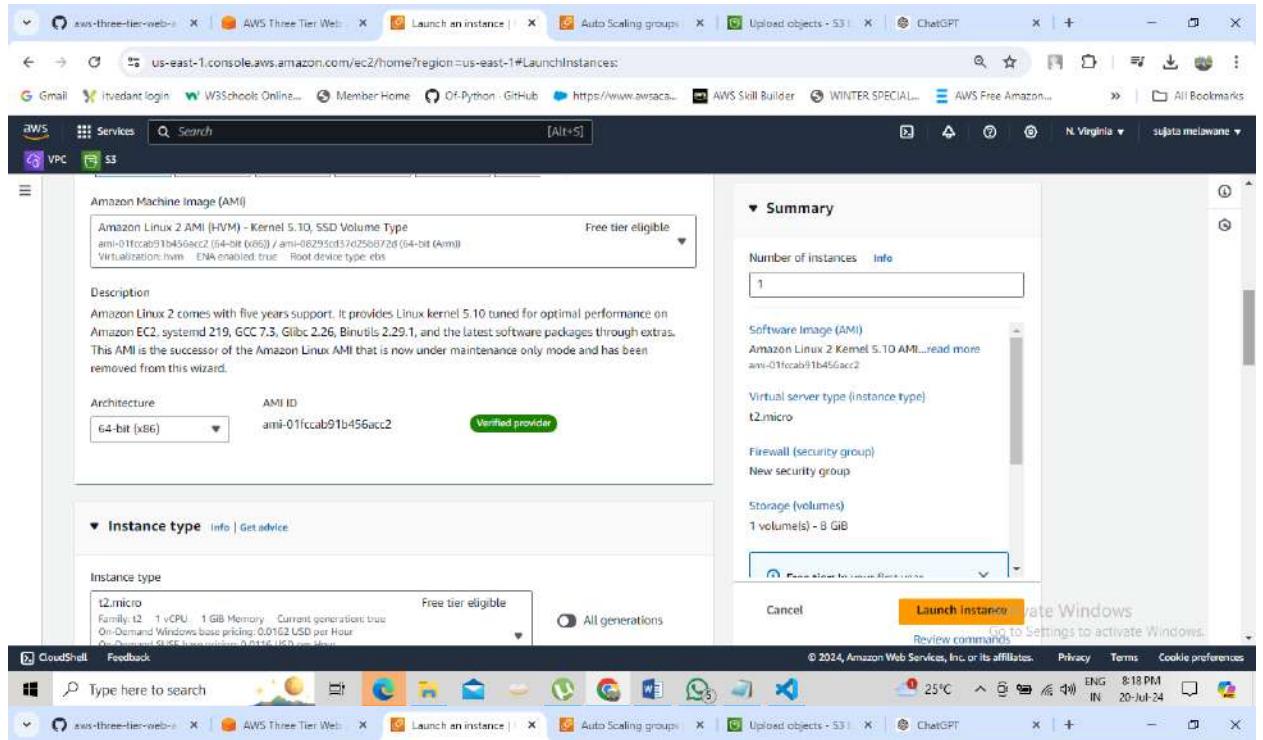
At the bottom, there is a status bar with the message "Activate Windows Go to Settings to activate Windows." and system information like "26°C", "20-07-24", and "8:08 PM".

# Web Instance Deployment

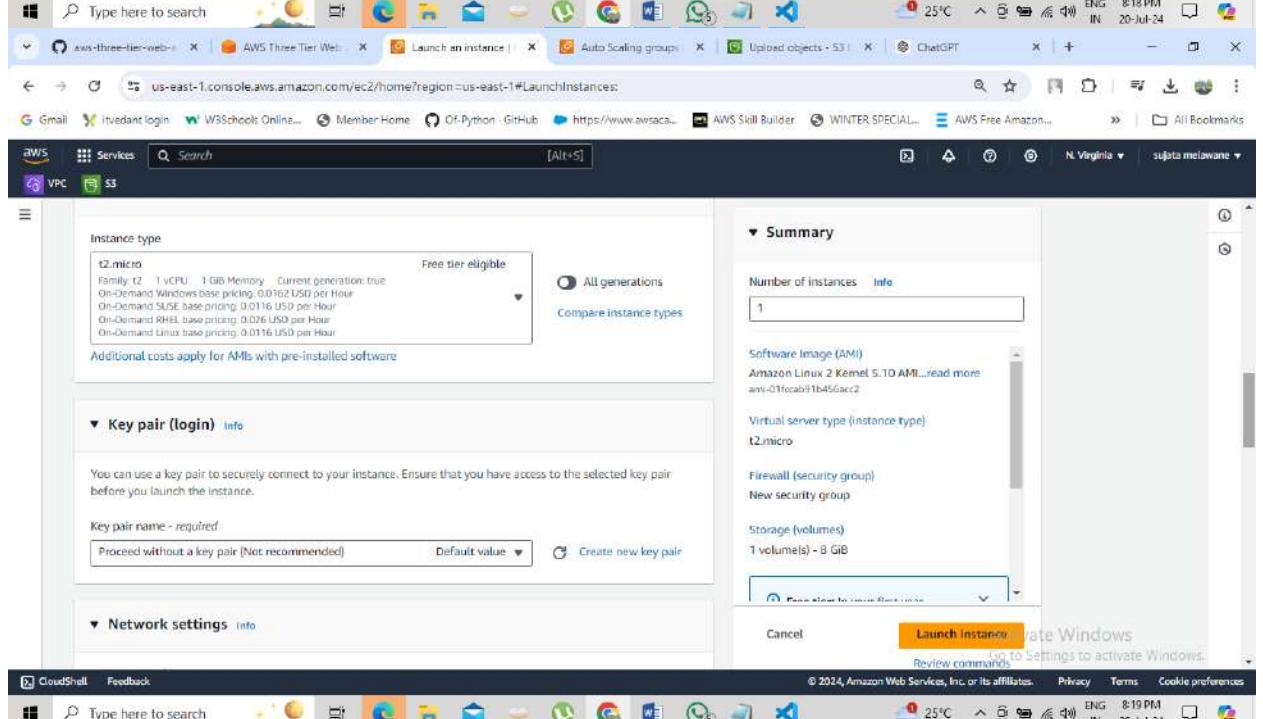
- Follow the same instance creation instructions we used for the App Tier instance in **Part 3: App Tier Instance Deployment**, with the exception of the subnet. We will be provisioning this instance in one of our **public subnets**. Make sure to select the correct network components, security group, and IAM role. **This time, auto-assign a public ip** on the **Configure Instance Details page**. Remember to tag the instance with a name so we can identify it more easily.



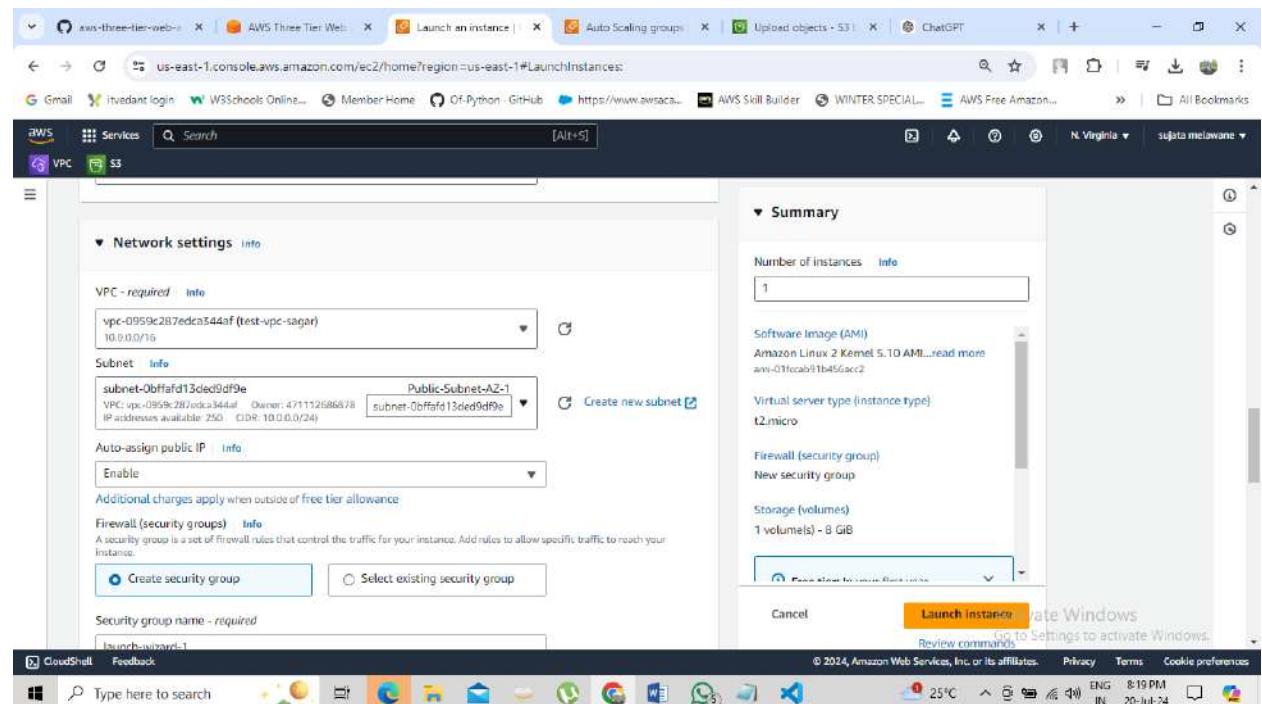
2.



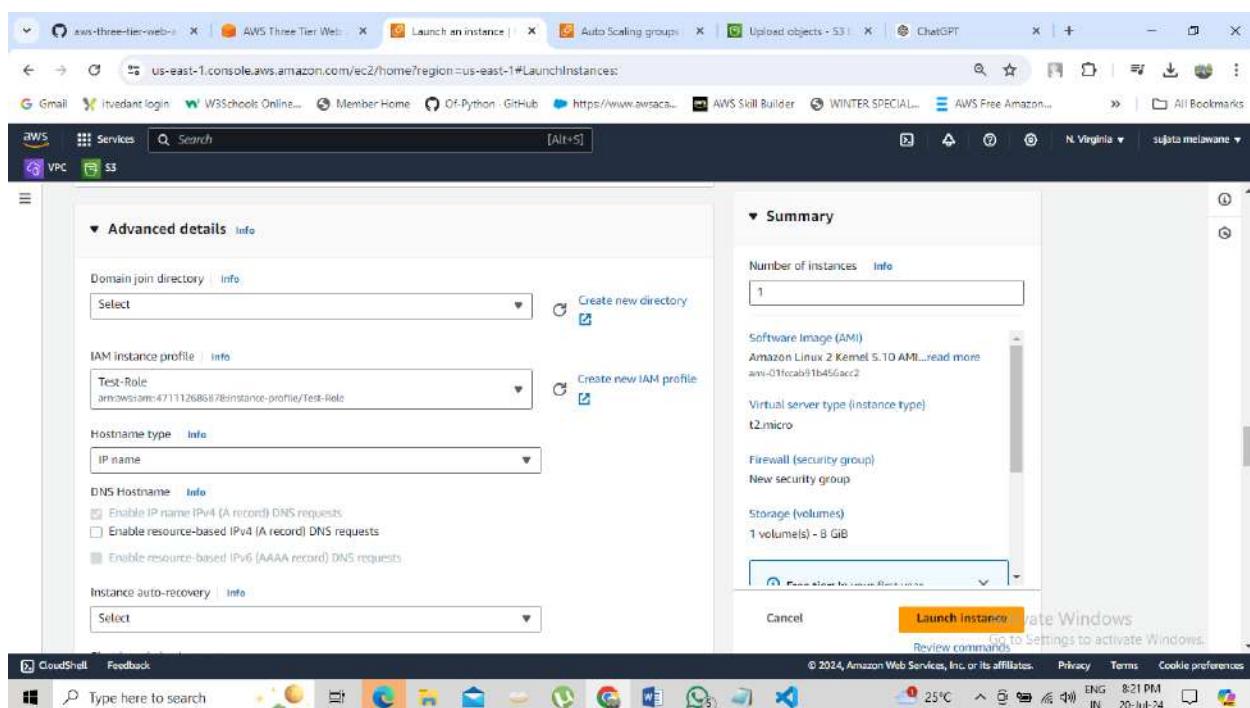
3.



4.



5.

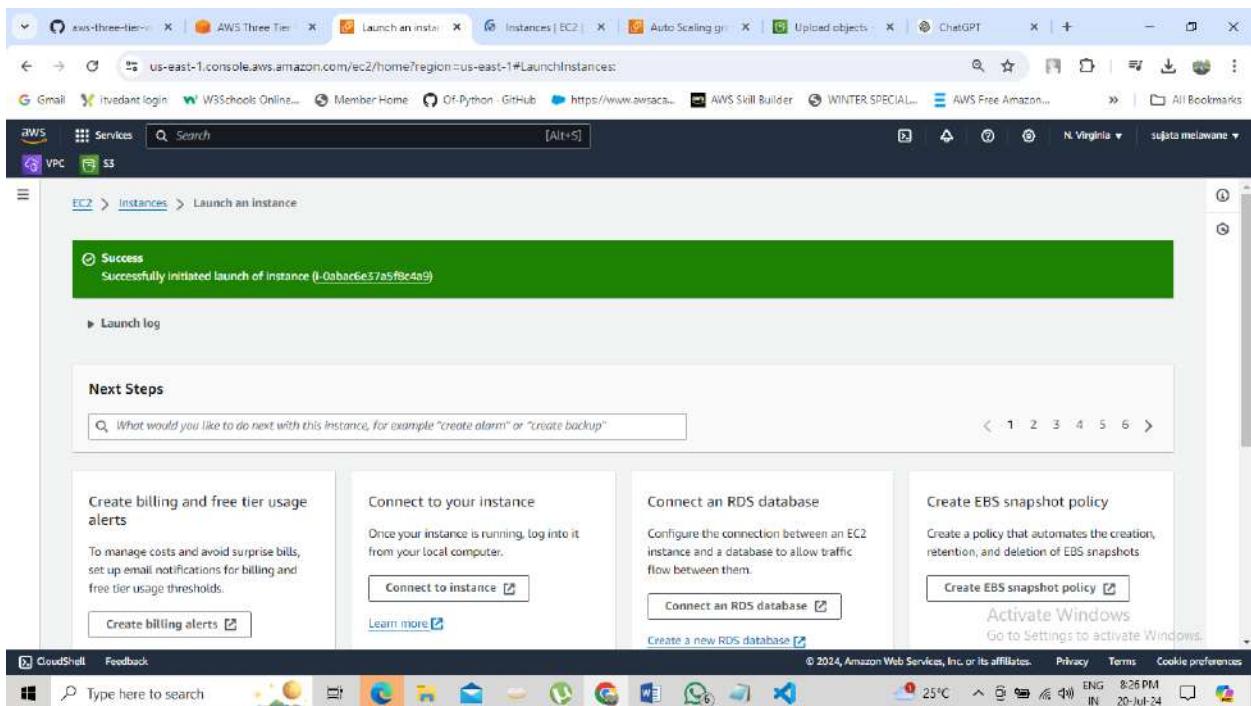


The screenshot shows the AWS CloudShell interface. A new EC2 instance is being configured. The instance type is t2.micro, and it is using the Web-Tier-SG security group. The storage is set to 1 volume(s) - 8 GiB. The 'Launch instance' button is highlighted.

The screenshot shows the AWS CloudShell interface. The 'Compare security group rules' section is active, comparing the rules of the selected security groups. The inbound rules table lists two entries for port 80:

| Security group name | Security group ID    | Type | Protocol | Port range | Source               | Description |
|---------------------|----------------------|------|----------|------------|----------------------|-------------|
| Web-Tier-SG         | sg-06216c771be90b77b | HTTP | tcp      | 80         | 114.79.138.90/32     | -           |
| Web-Tier-SG         | sg-06216c771be90b77b | HTTP | tcp      | 80         | sg-0f3990664fcefe796 | -           |

Then at the end, proceed without a key pair for this instance.



# Connect to Instance

1. Follow the same steps you used to connect to the app instance and change the user to **ec2-user**.  
Test connectivity here via ping as well since this instance should have internet connectivity:

```
1
2
sudo -su ec2-user
ping 8.8.8.8
```

*Note: If you don't see a transfer of packets then you'll need to verify your route tables attached to the subnet that your instance is deployed in*

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, and Instances. The Instances section is expanded, showing sub-options like Instances, Instance Types, Launch Templates, etc. The main pane displays a table of instances. A single instance is selected, with its details shown in a modal below:

| Name            | Instance ID         | Instance state | Instance type | Status check      | Alarm status | Availability Zone |
|-----------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|
| IAM-TEST-ROLE-2 | i-0abac6e37a5f8c4a9 | Running        | t2.micro      | 2/2 checks passed | View alarms  | us-east-1b        |

Below the table, the instance details are shown:

**i-0abac6e37a5f8c4a9 (IAM-TEST-ROLE-2)**

**Details** | Status and alarms | Monitoring | Security | Networking | Storage | Tags

**Instance summary**

| Instance ID                           | Public IPv4 address          | Private IPv4 addresses |
|---------------------------------------|------------------------------|------------------------|
| i-0abac6e37a5f8c4a9 (IAM-TEST-ROLE-2) | 3.227.241.155   open address | 10.0.0.180             |

IPv6 address: -  
Instance state: Running  
Private IP/DNS name (IPv4 only): -  
Public IPv4 DNS: -

Activate Windows  
Go to Settings to activate Windows.

The screenshot shows the AWS Systems Manager Session Manager terminal. The session ID is root-ckhhmwthig4ja4cmnsowioofq and the instance ID is i-0abac6e37a5f8c4a9. The terminal window displays the following command and output:

```
sh-4.2$ sudo -u ec2-user
[ec2-user@ip-10-0-0-180 bin]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=1.37 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=114 time=1.48 ms
```

At the bottom right of the terminal window, there's an "Activate Windows" message: "Go to Settings to activate Windows."

```

64 bytes from 8.8.8.8: icmp_seq=17 ttl=114 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=114 time=1.40 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=114 time=1.37 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=114 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=114 time=1.44 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=114 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=114 time=1.43 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=114 time=1.42 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=114 time=3.48 ms
64 bytes from 8.8.8.8: icmp_seq=26 ttl=114 time=1.42 ms
64 bytes from 8.8.8.8: icmp_seq=27 ttl=114 time=1.40 ms
64 bytes from 8.8.8.8: icmp_seq=28 ttl=114 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=29 ttl=114 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=30 ttl=114 time=1.36 ms
64 bytes from 8.8.8.8: icmp_seq=31 ttl=114 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=32 ttl=114 time=1.36 ms
64 bytes from 8.8.8.8: icmp_seq=33 ttl=114 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=34 ttl=114 time=1.36 ms
64 bytes from 8.8.8.8: icmp_seq=35 ttl=114 time=1.42 ms
64 bytes from 8.8.8.8: icmp_seq=36 ttl=114 time=1.40 ms
64 bytes from 8.8.8.8: icmp_seq=37 ttl=114 time=1.37 ms
64 bytes from 8.8.8.8: icmp_seq=38 ttl=114 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=39 ttl=114 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=40 ttl=114 time=1.34 ms
64 bytes from 8.8.8.8: icmp_seq=41 ttl=114 time=2.33 ms
64 bytes from 8.8.8.8: icmp_seq=42 ttl=114 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=43 ttl=114 time=1.38 ms
64 bytes from 8.8.8.8: icmp_seq=44 ttl=114 time=1.45 ms
*c
--- 8.8.8.8 ping statistics ---
44 packets transmitted, 44 received, 0% packet loss, time 43066ms
rtt min/avg/max/mdev = 1.348/1.486/3.482/0.347 ms
[ec2-user@ip-10-0-0-180 bin]$

```

Activate Windows  
Go to Settings to activate Windows.

*Note: If you don't see a transfer of packets then you'll need to verify your route tables attached to the subnet that your instance is deployed in.*

## Configure Web Instance

1. We now need to install all of the necessary components needed to run our front-end application. Again, start by installing NVM and node :

```

1
2
3
4
curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
nvm install 16
nvm use 16

```

2. Now we need to download our web tier code from our s3 bucket:

```
cd ~/
aws s3 cp s3://BUCKET_NAME/web-tier/ web-tier --recursive
```

Navigate to the web-layer folder and create the build folder for the react app so we can serve our code:

```
1
2
3
cd ~/web-tier
npm install
npm run build
```

```
Session ID: root-ckhhmwthig4ja4cmnsowioofsq Instance ID: i-0abac6e37a5fbc4a9#
Available packages:
36 BCC available [=0.x -stable]
37 mono available [=5.x -stable]
38 nginx=latest enabled [=stable]
40 mock available [=stable]
43 livepatch available [=stable]
44 tpython3.8 available [=stable]
45 haproxy2 available [=stable]
46 collectd available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10=latest enabled [=stable]
56 redis6 available [=stable]
58 tpostgresql12 available [=stable]
59 tpostgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 tpostgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
66 tphp8.1 available [=stable]
67 awsccli available [=stable]
68 tphp8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collectd-python3 available [=stable]
+ Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-10-0-0-180 ~]$
```

```
Session ID: root-ckhhmwthig4ja4cmnsowioofsq Instance ID: i-0abac6e37a5fbc4a9#
Available packages:
43 livepatch available [=stable]
44 tpython3.8 available [=stable]
45 haproxy2 available [=stable]
46 collectd available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10=latest enabled [=stable]
56 redis6 available [=stable]
58 tpostgresql12 available [=stable]
59 tpostgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 tpostgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
66 tphp8.1 available [=stable]
67 awsccli available [=stable]
68 tphp8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collectd-python3 available [=stable]
+ Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-10-0-0-180 ~]$ cd /etc/nginx
[ec2-user@ip-10-0-0-180 nginx]$ ls
conf.d fastcgi.conf fastcgi_params koi-utf mime.types nginx.conf scgi_params uwsgi_params
error.log fastcgi.conf.default fastcgi_params.default koi-win mime.types.default nginx.conf.default scgi_params.default uwsgi_params.default
[ec2-user@ip-10-0-0-180 nginx]$
```

```
Session ID: root-ckhhmwthig4ja4cmnsowioofsq Instance ID: i-0abac6e37a5fbc4a9#
Available packages:
43 livepatch available [=stable]
44 tpython3.8 available [=stable]
45 haproxy2 available [=stable]
46 collectd available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10=latest enabled [=stable]
56 redis6 available [=stable]
58 tpostgresql12 available [=stable]
59 tpostgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 tpostgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
66 tphp8.1 available [=stable]
67 awsccli available [=stable]
68 tphp8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collectd-python3 available [=stable]
+ Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-10-0-0-180 ~]$ cd /etc/nginx
[ec2-user@ip-10-0-0-180 nginx]$ ls
conf.d fastcgi.conf fastcgi_params koi-utf mime.types nginx.conf scgi_params uwsgi_params
error.log fastcgi.conf.default fastcgi_params.default koi-win mime.types.default nginx.conf.default scgi_params.default uwsgi_params.default
[ec2-user@ip-10-0-0-180 nginx]$
```

```
aws-three-1 | AWS Three | Systems M... | Instances | Auto Scaling | Test-Sagar | ChatGPT | 3.227.241.1 | - | + | X
us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-0abac6e37a5f8c4a9?region=us-east-1#;

Session ID: root-hipx66606vxxfsut3pomq3duu Instance ID: i-0abac6e37a5f8c4a9
Terminate

download: s3://test-sagar-s3/web-tier/src/assets/3TierArch.png to web-tier/src/assets/3TierArch.png
download: s3://test-sagar-s3/web-tier/src/App.css to web-tier/src/App.css
download: s3://test-sagar-s3/web-tier/src/setupTests.js to web-tier/src/setupTests.js
download: s3://test-sagar-s3/web-tier/src/theme.js to web-tier/src/theme.js
download: s3://test-sagar-s3/web-tier/src/index.css to web-tier/src/index.css
download: s3://test-sagar-s3/web-tier/src/reportWebVitals.js to web-tier/src/reportWebVitals.js
download: s3://test-sagar-s3/web-tier/public/robots.txt to web-tier/public/robots.txt
download: s3://test-sagar-s3/web-tier/src/App.js to web-tier/src/App.js
download: s3://test-sagar-s3/web-tier/src/components/iDatabaseDemo/DatabaseDemo.js to web-tier/src/components/DatabaseDemo/DatabaseDemo.js
[ec2-user@ip-10-0-0-180 ~] cd ./web-tier
[ec2-user@ip-10-0-0-180 web-tier]$ npm install
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /home/ec2-user/web-tier/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/home/ec2-user/web-tier/package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

A complete log of this run can be found in:
npm ERR! /home/ec2-user/.npm/_logs/2024-07-20T18_32_36_606z-debug-0.log
[ec2-user@ip-10-0-0-180 web-tier]$ npm run build
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /home/ec2-user/web-tier/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/home/ec2-user/web-tier/package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

A complete log of this run can be found in:
npm ERR! /home/ec2-user/.npm/_logs/2024-07-20T18_32_37_3622-debug-0.log
[ec2-user@ip-10-0-0-180 web-tier]$

Activate Windows
Go to Settings to activate Windows.
25°C Cloudy ENG 12:02 AM IN 21-Jul-24
```

```
aws-three-1 | AWS Three | Systems M... | Instances | Auto Scaling | Test-Sagar | ChatGPT | 3.227.241.1 | - | + | X
us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-0abac6e37a5f8c4a9?region=us-east-1#;

Session ID: root-hipx66606vxxfsut3pomq3duu Instance ID: i-0abac6e37a5f8c4a9
Terminate

36 BCC available [=0.x =stable]
37 mono available [=5.x =stable]
38 nginx@latest enabled [=stable]
39 mock available [=stable]
40 livepatch available [=stable]
41 tpython3.9 available [=stable]
45 haproxy2 available [=stable]
46 collected available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
50 kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 context9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10@latest enabled [=stable]
56 zedis6 available [=stable]
58 psycopgql12 available [=stable]
59 psycopgql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.16 available [=stable]
63 psycopgql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
66 tphp8.1 available [=stable]
67 asecli available [=stable]
68 tphp8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collected-python3 available [=stable]
+ Note: End-of-support. Use 'info' subcommand.
[ec2-user@ip-10-0-0-180 web-tier]$

Activate Windows
Go to Settings to activate Windows.
25°C Cloudy ENG 12:03 AM IN 21-Jul-24
```



