

Project By:

S. GUHAN BALAJI



AWS 3 Tier Architecture:

AWS 3-tier architecture is a cloud computing architecture that divides an application into three logical tiers:

- Presentation tier: This tier is responsible for interacting with the end user. It typically consists of web servers that deliver HTML pages, images, and other content to the user's browser.
- Application tier: This tier is responsible for processing business logic and data. It typically consists of application servers that run the application's code and interact with the database.
- Data tier: This tier stores the application's data. It typically consists of a database server that manages the data and provides access to it.

The three tiers are typically separated into different Amazon Web Services (AWS) services. For example, the presentation tier might be hosted on Elastic Load Balancing (ELB) and Amazon EC2 instances, the application tier might be hosted on Amazon EC2 instances, and the data tier might be hosted on Amazon Relational Database Service (RDS).

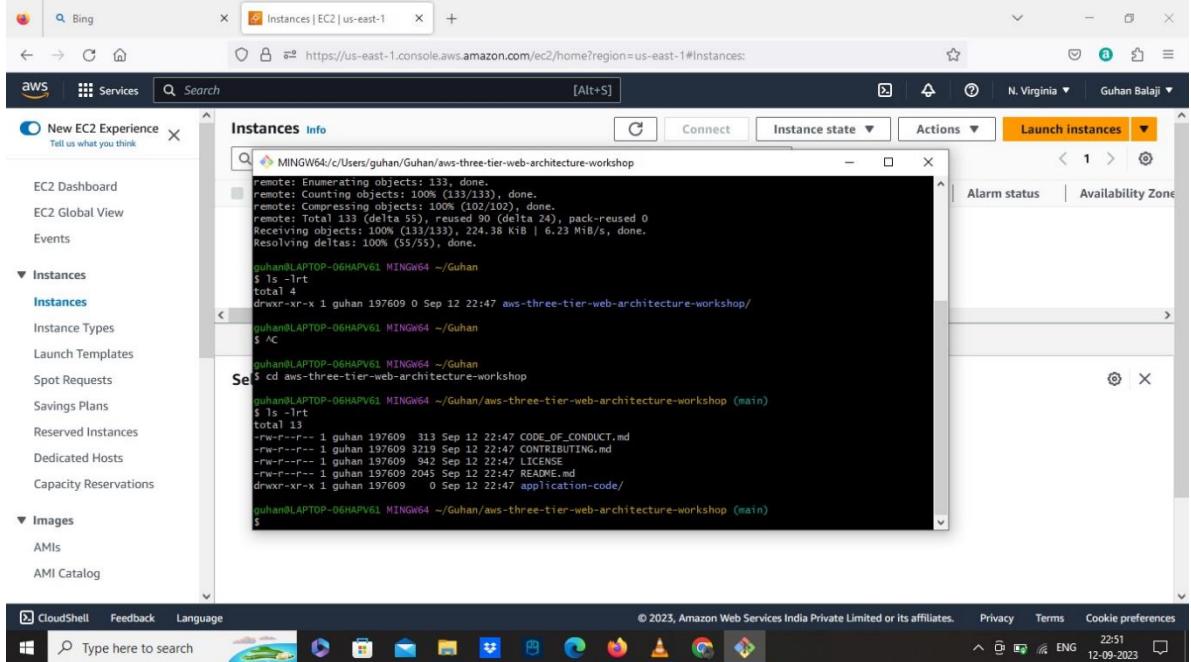
AWS 3-tier architecture offers a number of benefits, including:

- Scalability: The application can be scaled up or down easily to meet changing demand.
- Availability: The application can be made highly available by deploying the tiers in multiple Availability Zones.
- Security: The tiers can be isolated from each other to improve security.
- Cost-effectiveness: The application can be deployed in a cost-effective way by using the right AWS services.

AWS 3-tier architecture is a popular choice for building scalable, reliable, and secure applications. It is a good choice for a variety of applications, including web applications, mobile applications, and enterprise applications.

Steps involved:

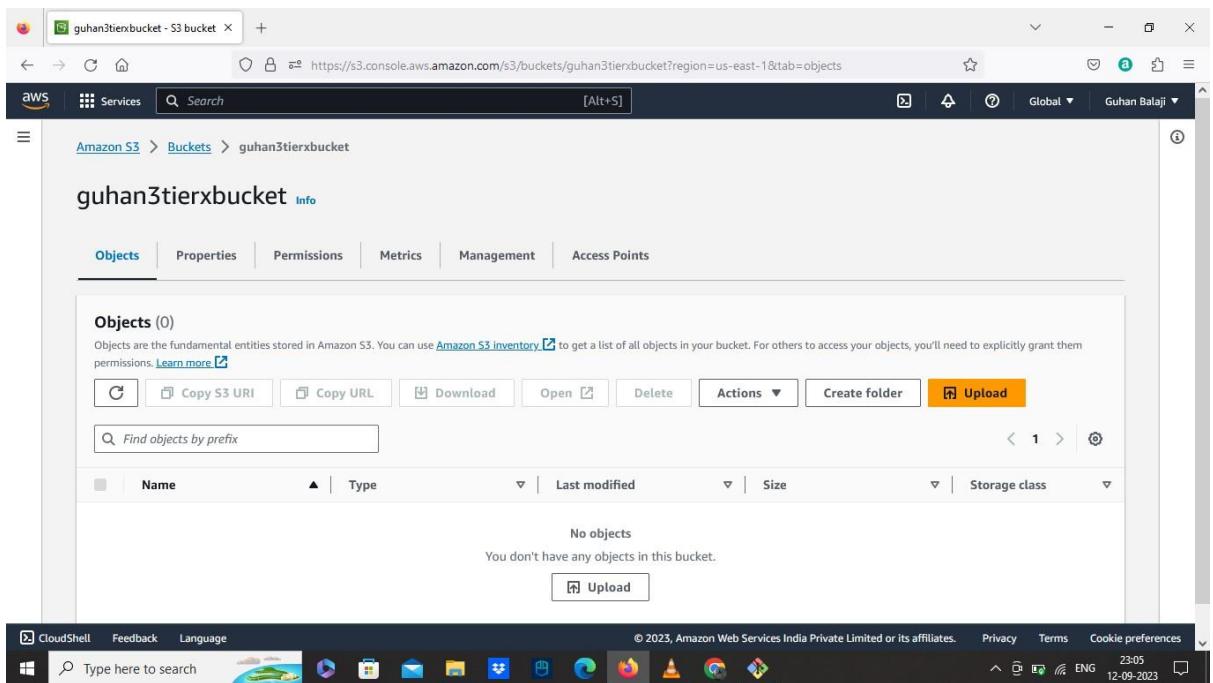
- Download the App tier, Web tier files from the Github Using Gitbash to your computer.



```
remote: Enumerating objects: 133, done.
remote: Counting objects: 100% (133/133), done.
remote: Compressing objects: 100% (102/102), done.
remote: Total 133 (delta 55), reused 90 (delta 24), pack-reused 0
Receiving objects: 100% (133/133), 224.38 KiB | 6.23 MiB/s, done.
Resolving deltas: 100% (55/55), done.

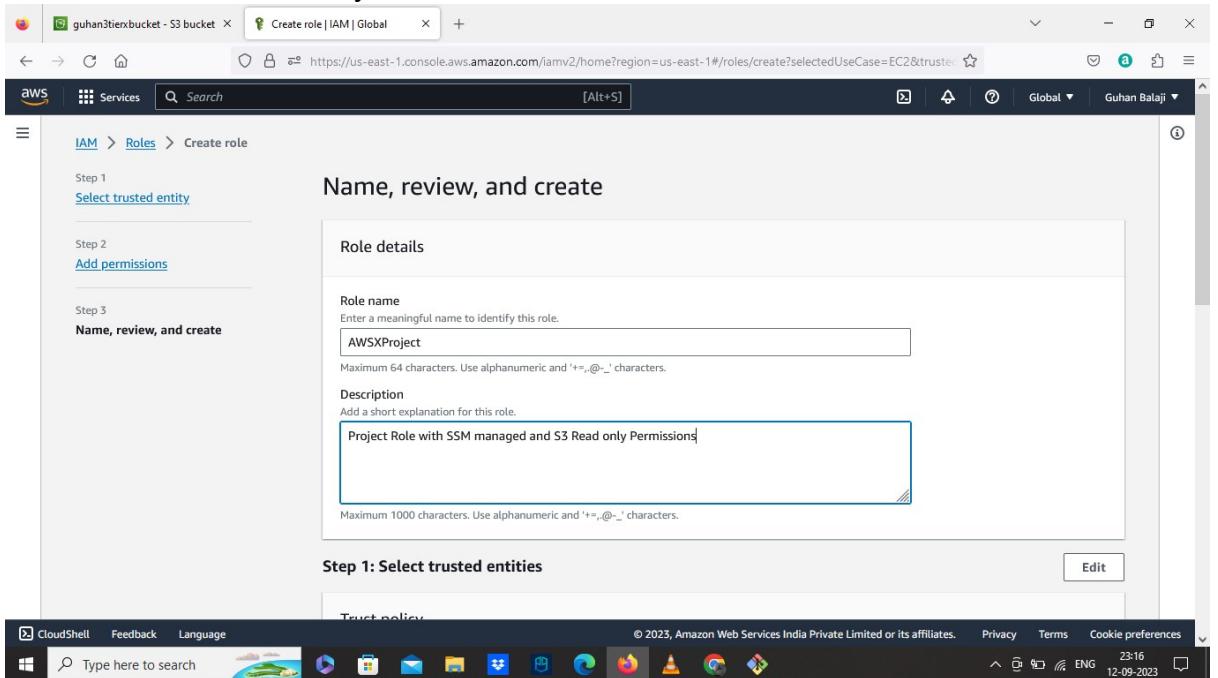
guhan@LAPTOP-06HAPV61 MINGW64 ~/Guhan
$ ls -lrt
total 4
drwxr-xr-x 1 guhan 197609 0 Sep 12 22:47 aws-three-tier-web-architecture-workshop/
guhan@LAPTOP-06HAPV61 MINGW64 ~/Guhan
$ cd aws-three-tier-web-architecture-workshop
guhan@LAPTOP-06HAPV61 MINGW64 ~/Guhan/aws-three-tier-web-architecture-workshop (main)
$ ls -lrt
total 13
drwxr-xr-x 1 guhan 197609 313 Sep 12 22:47 CODE_OF_CONDUCT.md
-rw-r--r-- 1 guhan 197609 3219 Sep 12 22:47 CONTRIBUTING.md
-rw-r--r-- 1 guhan 197609 942 Sep 12 22:47 LICENSE
-rw-r--r-- 1 guhan 197609 2045 Sep 12 22:47 README.md
drwxr-xr-x 1 guhan 197609 0 Sep 12 22:47 application-code/
guhan@LAPTOP-06HAPV61 MINGW64 ~/Guhan/aws-three-tier-web-architecture-workshop (main)
```

- Create a new S3 bucket in your AWS console.

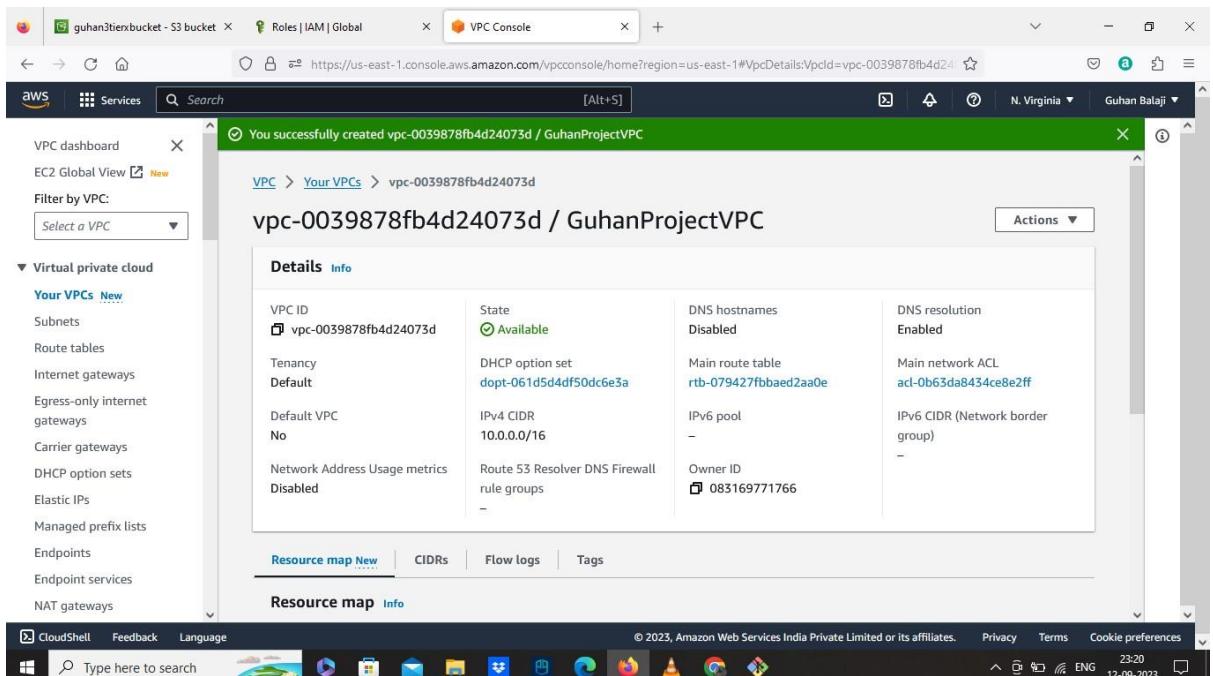


- Create a new IAM role:
 - Select AWS services as Trusted entity type.
 - Select EC2 for common use case.

- Add permissions to your role AmazonSSMmanagedInstanceCore and Amazon S3 read only access.



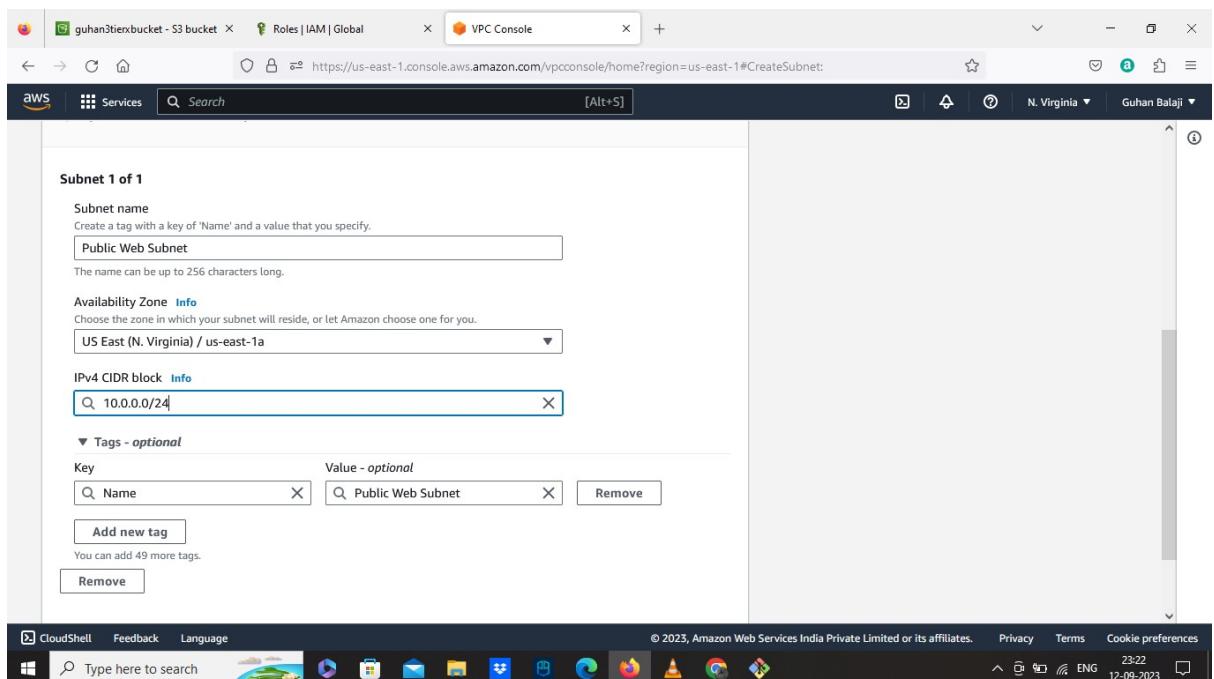
- Create a Custom VPC:
 - Select VPC only
 - Select IPV4 CIDR manual
 - Use 10.0.0.0/16 CIDR block
 - No IPV6



- Create Subnets
Here we need to create 6 subnets in two different regions for each tier.

For WEB TIER: Two Public Subnets

- Subnet 1:
 - ✓ Add our created VPC ID
 - ✓ Name as Public Web Subnet AZ1
 - ✓ Select Availability zone- US east 1A
 - ✓ Select 10.0.0.0/24 as IPV4 CIDR block



- Subnet 2:
 - ✓ Add our created VPC ID
 - ✓ Name as Public Web Subnet AZ2
 - ✓ Select Availability zone- US east 1B
 - ✓ Select 10.0.1.0/24 as IPV4 CIDR block

For APP TIER: Two Private Subnets

- Subnet 3:
 - ✓ Add our created VPC ID
 - ✓ Name as Private App Subnet AZ1
 - ✓ Select Availability zone- US east 1a.
 - ✓ Select 10.0.2.0/24 as IPV4 CIDR block
- Subnet 4:
 - ✓ Add our created VPC ID
 - ✓ Name as Private App Subnet AZ2
 - ✓ Select Availability zone- US east 1B
 - ✓ Select 10.0.3.0/24 as IPV4 CIDR block

For DB TIER: Two Private Subnets

- Subnet 5:
 - ✓ Add our created VPC ID
 - ✓ Name as Private DB Subnet AZ1
 - ✓ Select Availability zone- US east 1a
 - ✓ Select 10.0.4.0/24 as IPV4 CIDR block
- Subnet 6:
 - ✓ Add our created VPC ID
 - ✓ Name as Private DB Subnet AZ2
 - ✓ Select Availability zone- US east 1b
 - ✓ Select 10.0.5.0/24 as IPV4 CIDR block.

The screenshot shows the AWS VPC Console interface. A modal window in the center says "You have successfully created 1 subnet: subnet-0dc13548d35bac750". Below it is a table titled "Subnets (12) Info" with the following data:

Name	Subnet ID	State	VPC	IPv4
-	subnet-099bdababdb8ac3b5	Available	vpc-03e358b2345c99074	172.31.1.0/24
-	subnet-02b435da27002bee3	Available	vpc-03e358b2345c99074	172.31.2.0/24
Private DB Subnet AZ -1	subnet-043105e5fffffdf05	Available	vpc-0039878fb4d24073d Guhan's VPC	10.0.1.0/24
Public Web Subnet	subnet-08734531433d9c964	Available	vpc-0039878fb4d24073d Guhan's VPC	10.0.2.0/24
-	subnet-046d1ea89eeb9de05	Available	vpc-03e358b2345c99074	172.31.3.0/24
Public Web Subnet AZ-2	subnet-025bece6e11527ae5	Available	vpc-0039878fb4d24073d Guhan's VPC	10.0.3.0/24
Private App Subnet AZ - 2	subnet-05375332dd95c0d3c	Available	vpc-0039878fb4d24073d Guhan's VPC	10.0.4.0/24
Private App Subnet AZ-1	subnet-0d3cc9fa0ef9e5175	Available	vpc-0039878fb4d24073d Guhan's VPC	10.0.5.0/24
Private DB Subnet AZ - 2	subnet-0dc13548d35bac750	Available	vpc-0039878fb4d24073d Guhan's VPC	10.0.6.0/24

- Create Internet Gateway:

- Attach to our Custom VPC.

The screenshot shows the AWS VPC Console interface. The main content area displays a confirmation message: "Internet gateway igw-065dcf4142046b6f2 successfully attached to vpc-0039878fb4d24073d". Below this, the title "igw-065dcf4142046b6f2 / Guhan Project 3 tier" is shown. The "Details" tab is selected, displaying the following information:

Internet gateway ID igw-065dcf4142046b6f2	State Attached	VPC ID vpc-0039878fb4d24073d GuhanProjectVPC	Owner 083169771766
--	-------------------	--	-----------------------

Below the details, there is a "Tags" section with one entry: "Name" set to "Guhan Project 3 tier". The left sidebar shows the navigation menu for the VPC service, including options like "Virtual private cloud", "Your VPCs", "Subnets", "Route tables", "Internet gateways", and "NAT gateways". The "Internet gateways" option is currently selected. The bottom of the screen shows the Windows taskbar with various pinned icons.

- Create Two NAT Gateways
 - One for one availability zone 1a and other for 1b. Allocate Elastic IP for each.

The screenshot shows the AWS VPC Console interface. The main content area displays a confirmation message: "NAT gateway nat-0128e8bfd989ae486 | AZ1 NAT gateway was created successfully.". Below this, the title "nat-0128e8bfd989ae486 / AZ1 NAT gateway" is shown. The "Details" tab is selected, displaying the following information:

NAT gateway ID nat-0128e8bfd989ae486	Connectivity type Public	State Pending	State message -
NAT gateway ARN arn:aws:ec2:us-east-1:083169771766:natgateway/nat-0128e8bfd989ae486	Primary public IPv4 address -	Primary private IPv4 address 10.0.0.7	Primary network interface ID eni-029cdc05191ae32fc
VPC vpc-0039878fb4d24073d / GuhanProjectVPC	Subnet subnet-08734531433d9c964 / Public Web Subnet	Created Tuesday, September 12, 2023 at 23:39:11 GMT+5:30	Deleted -

Below the details, there are tabs for "Secondary IPv4 addresses", "Monitoring", and "Tags". The left sidebar shows the navigation menu for the VPC service, including options like "Virtual private cloud", "Your VPCs", "Subnets", "Route tables", "Internet gateways", and "NAT gateways". The "NAT gateways" option is currently selected. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the AWS VPC Console with a success message: "NAT gateway nat-06be1a5e0863be7a1 | AZ2 NAT gateway was created successfully." The main details pane shows the following information for the NAT gateway:

NAT gateway ID	Connectivity type	State	State message
nat-06be1a5e0863be7a1	Public	Pending	-
NAT gateway ARN	Primary public IPv4 address	Primary private IPv4 address	Primary network interface ID
arn:aws:ec2:us-east-1:083169771766:natgateway/nat-06be1a5e0863be7a1	-	10.0.1.28	eni-085f454466030df02
VPC	Subnet	Created	Deleted
vpc-0039878fb4d24073d / GuhanProjectVPC	subnet-025bece6e11527ae5 / Public Web Subnet AZ-2	Tuesday, September 12, 2023 at 23:40:31 GMT+5:30	-

The sidebar on the left lists various VPC components like Route tables, Internet gateways, and NAT gateways. The bottom navigation bar includes CloudShell, Feedback, Language, and other AWS services.

- Create Two Route Table:
 - Route Table 1:

- ✓ Select our VPC.
- ✓ After created, Go to Edit Routes and add Route destination as 0.0.0.0/0 and select target as Internet Gateway we created.
- ✓ Then go to subnet association and edit subnet association. Select two Public Subnets AZ1,AZ2.

The screenshot shows the AWS VPC Console with a success message: "Updated routes for rtb-05161cf4f29f1b48d / Public Route Table successfully". The main routes table shows two entries:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-065dcf4142046b6f2	Active	No
10.0.0.0/16	local	Active	No

The sidebar on the left lists various VPC components like Route tables, Internet gateways, and NAT gateways. The bottom navigation bar includes CloudShell, Feedback, Language, and other AWS services.

The screenshot shows the AWS VPC Console interface. A success message at the top states: "You have successfully updated subnet associations for rtb-05161cf4f29f1b48d / Public Route Table." Below this, the "Details" tab is selected in a card view. The card displays information about the route table, including its ID (rtb-05161cf4f29f1b48d), being the "Main" route table (No), and having "Explicit subnet associations" (2 subnets). The "Subnet associations" tab is active, showing two explicit subnet associations: "Public Web Subnet" (subnet-08734531433d9c964) with IPv4 CIDR 10.0.0.0/24 and "Public Web Subnet AZ-2" (subnet-025bece6e11527ae5) with IPv4 CIDR 10.0.1.0/24. There are also four subnets listed under "Subnets without explicit associations".

- **Route Table 2: (Private Route Table)**
 - ✓ Select our VPC.
 - ✓ Then edit routes, add route destination as 0.0.0.0/0 and Select target as NAT gateway we created.
 - ✓ Go to subnet association and select two Private App Subnets AZ1,AZ2.

The screenshot shows the AWS VPC Console interface. The "Route tables" section lists four route tables: "Public Route Table" (rtb-05161cf4f29f1b48d, Main, 2 subnets), "Private Route Table" (rtb-0275c89f26e4206b8, Main, 1 subnet), and two others. The "Private Route Table" is selected. Below it, the "rtb-0720f25f6af93fef2 / Private Route Table AZ2" details page is shown. This page includes tabs for "Details", "Routes", "Subnet associations", "Edge associations", "Route propagation", and "Tags". The "Details" tab is selected, showing the route table ID (rtb-0720f25f6af93fef2), being the "Main" route table (No), and having no explicit subnet associations or edge associations.

- **Create Security Group**
Here we need to create 5 security groups which is for Internet Facing ELB, Web tier, ILB, Private Instances, Databases.

- SG 1 (for Internet facing ELB)
 - ✓ Select our VPC
 - ✓ Add inbound rule1 – Type as HTTP, Source as 0.0.0.0/0 and port range as 80
 - ✓ Add inbound rule 2 – Select Type as custom TCP and select Anywhere IPV6.
 - ✓ And Create.

The screenshot shows the AWS VPC Console interface for creating a new security group. The 'Basic details' section includes:

- Security group name:** External Load Balancer SG
- Description:** External Load Balancer
- VPC:** vpc-0039878fb4d24073d

The 'Inbound rules' section contains one rule:

Type Info	Protocol	Port range Info	Source Info	Description - optional info
HTTP	TCP	80	Any... 0.0.0.0/0	

- SG 2 (For Web Tier)
 - ✓ Select our VPC

The screenshot shows the AWS VPC Console interface for creating a new security group. The 'Basic details' section includes:

- Security group name:** Web Tier SG
- Description:** For WEB Tier
- VPC:** vpc-0039878fb4d24073d

The 'Inbound rules' section is currently empty.

- ✓ Add Inbound Rule 1 – Select HTTP, source as External LB SG (previously created SG)
- ✓ Add inbound rule 2 – Select HTTP, source as MY IP.
- SG3 (Internal Load Balancer)
 - ✓ Select VPC
 - ✓ Add Inbound Rule 1 – Select HTTP, Source as Web Tier SG

Security group (sg-034231f116aa3befb | Internal Load balancer SG) was created successfully

sg-034231f116aa3befb - Internal Load balancer SG

Details			
Security group name Internal Load balancer SG	Security group ID sg-034231f116aa3befb	Description Internal Load Balancer SG	VPC ID vpc-0039878fb4d24073d
Owner 083169771766	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules (1/1)

- SG4 (Private Instances)
 - ✓ Select our VPC
 - ✓ Add inbound rule 1 – Port range – 4000,source as Internal Load Balancer SG

Inbound rules info

Type	Protocol	Port range	Source	Description
Custom TCP	TCP	4000	Cus... sg-034231f116aa3befb	
Custom TCP	TCP	4000	My IP 182.65.77.129/32	

Add rule

Outbound rules info

Type	Protocol	Port range	Destination	Description
------	----------	------------	-------------	-------------

- ✓ Add inbound rule 2 – Port Range as 4000, Source as MY IP
- SG5 (Database)
 - ✓ Select our VPC.
 - ✓ Add inbound rule 1 - Type as MY SQL/Aurora, Port as 3306, Source type as Private Instance SG.

Description Info
Database SG

VPC Info
vpc-0039878fb4d24073d

Inbound rules Info

Type Info	Protocol	Port range Info	Source Info	Description - optional Info
MySQL/Aurora	TCP	3306	Cus... sg-09cb108e44cf349d2	

Add rule

Outbound rules Info

Name	Security group ID	Security group name	VPC ID	Description
sg-03e5b37c0aaefee0f	Database SG	vpc-0039878fb4d24073d	Database SG	
sg-05795d7be5a7085aa	default	vpc-0039878fb4d24073d	default VPC security gr...	
sg-034231f116aa3befb	Internal Load balancer ...	vpc-0039878fb4d24073d	Internal Load Balancer...	
sg-09cb108e44cf349d2	Private SG	vpc-0039878fb4d24073d	Private App SG	
sg-0b7851227202ca53b	default	vpc-03e358b2345c99074	default VPC security gr...	
sg-0fe2f667558107d41	External Load Balancer...	vpc-0039878fb4d24073d	External Load Balancer...	
sg-0996b0a446a6dd6c3	Web Tier SG	vpc-0039878fb4d24073d	For WEB Tier	

- Create Database in Amazon RDS
 - Go to Subnet Groups
 - ✓ Create DB subnet groups

- ✓ Select our custom VPC
- ✓ Choose Availability Zone as 1a,1b
- ✓ Select the two database private subnets AZ1,AZ2

3 tier Subnet Groups

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

GuhanProjectVPC (vpc-0039878fb4d24073d)

Add subnets

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

Choose an availability zone
us-east-1a X us-east-1b X

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

Select subnets
subnet-0dc13548d35bac750 (10.0.5.0/24) X
subnet-043105e5fffffdf05 (10.0.4.0/24) X

Successfully created Guhan 3 tier. View subnet group

RDS > Subnet groups

Name	Description	Status	VPC
guhan 3 tier	3 tier Subnet Groups	Complete	vpc-0039878fb4d24073d

- Go to Databases
 - ✓ Choose Database creation method as Standard Create
 - ✓ Select Engine Type as AURORA/MYSQL
 - ✓ Choose Templates as DEV TEST
 - ✓ Create Master Username and Password
 - ✓ Select Create An Aurora Replica in availability and durability

- ✓ Select Don't Connect to an EC2 in Connectivity.
- ✓ Select Our VPC
- ✓ Select The Subnet Group we created
- ✓ Set Public Access as NO
- ✓ Select Existing Security group and choose Database SG
- ✓ Choose password authentication in Database authentication
- ✓ And finally Turn OFF the performance insights and create Database.

Create database

Choose a database creation method [Info](#)

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible)

Aurora (PostgreSQL Compatible)

MySQL

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Aurora (MySQL 5.7) 2.11.2

Templates
Choose a sample template to meet your use case.

Production
Use defaults for high availability and fast, consistent performance.

Dev/Test
This instance is intended for development use outside of a production environment.

Settings

DB cluster identifier [Info](#)
Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS RDS console with the URL <https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance>. The page is titled 'RDS | us-east-1'. The 'Services' tab is selected. The main section is titled 'Credentials Settings'. Under 'Master username', the value 'Guhan29' is entered into a text input field. A note states: 'Type a login ID for the master user of your DB instance.' Below this, there's a note about managing master credentials in AWS Secrets Manager, with a link to learn more. There's also an option to 'Auto generate a password'. The 'Master password' and 'Confirm master password' fields both contain masked text. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the AWS RDS console with the same URL as the previous screenshot. The page is titled 'RDS | us-east-1'. The 'Services' tab is selected. The main section is titled 'Connectivity'. It includes options for 'Compute resource': 'Don't connect to an EC2 compute resource' (selected) and 'Connect to an EC2 compute resource'. It also includes options for 'Network type': 'IPv4' (selected) and 'Dual-stack mode'. Under 'Virtual private cloud (VPC)', the value 'GuhanProjectVPC (vpc-0039878fb4d24073d)' is selected from a dropdown menu. The bottom of the screen shows the Windows taskbar with various pinned icons.

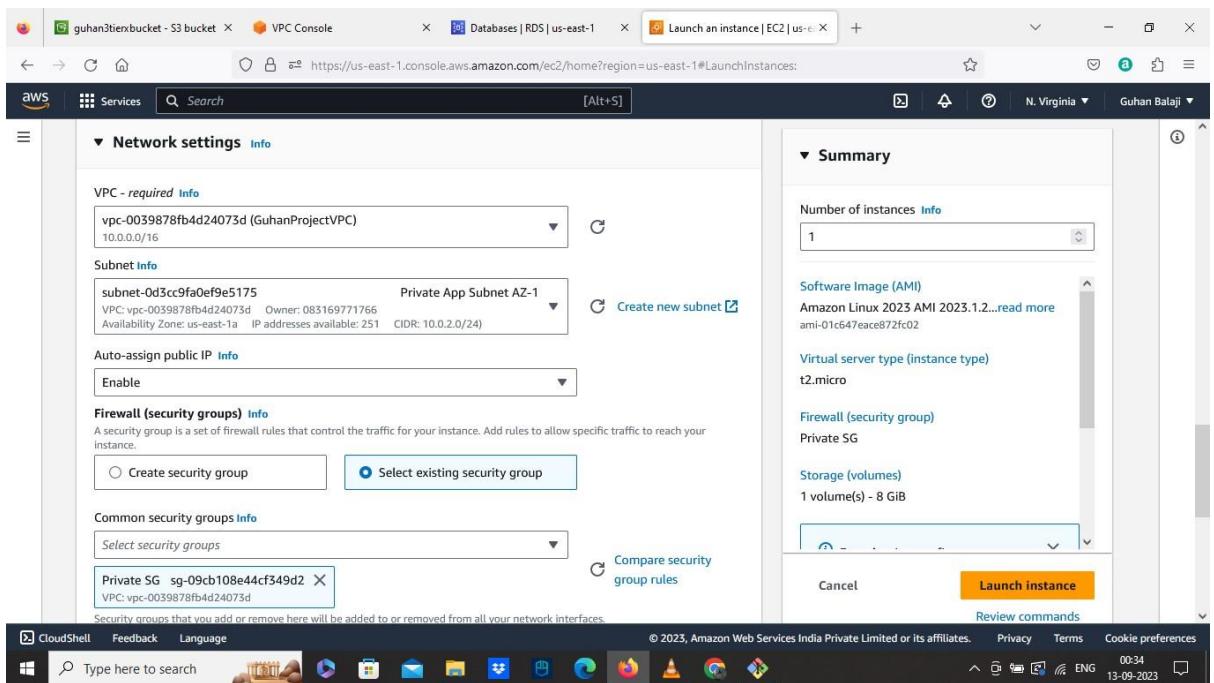
The screenshot shows the AWS RDS console for launching a new database instance. The 'VPC' tab is selected. A note at the top says: "To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify." Two radio button options are shown: "IPv4" (selected) and "Dual-stack mode". Below this, under "Virtual private cloud (VPC)", it says "Choose the VPC. The VPC defines the virtual networking environment for this DB cluster." A dropdown menu shows "GuhanProjectVPC (vpc-0039878fb4d24073d)" with "6 Subnets, 2 Availability Zones". A note below says "Only VPCs with a corresponding DB subnet group are listed." A callout box states: "After a database is created, you can't change its VPC." Under "DB subnet group", it says "Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected." A dropdown menu shows "guhan 3 tier" with "2 Subnets, 2 Availability Zones". Under "Public access", the "No" option is selected. A note says: "RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster." The bottom of the screen shows the Windows taskbar with various pinned icons.

This screenshot continues from the previous one, showing the RDS launch wizard. The "Public access" section is now visible, with the "No" option selected. A note explains: "RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster." Below this, the "VPC security group (firewall)" section is shown, with the "Choose existing" option selected. A dropdown menu lists "Choose existing VPC security groups" and "Create new VPC security group". Under "Existing VPC security groups", a single entry "Database SG" is listed. At the bottom, the "RDS Proxy" section is visible, with a note about its benefits and a checkbox for "Create an RDS Proxy". The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the AWS RDS console in the us-east-1 region. A new DB instance is being created with the identifier 'database-1'. The configuration includes an Aurora MySQL engine and a db.r6g.2xlarge instance type. The estimated monthly cost is listed as 757.74 USD. A note states that billing is based on on-demand usage and does not consider reserved instance benefits. A warning message indicates responsibility for ensuring necessary rights for third-party services. The 'Create database' button is visible at the bottom right.

The screenshot shows the AWS RDS console in the us-east-1 region, displaying the list of databases under the 'Databases' tab. The sidebar navigation includes 'Dashboard', 'Databases' (which is selected), 'Query Editor', 'Performance insights', 'Schemas', 'Exports in Amazon S3', 'Automated backups', 'Reserved instances', 'Proxies', 'Subnet groups', 'Parameter groups', 'Option groups', 'Custom engine versions', and 'Zero-ETL integrations'. The main content area shows three databases: 'database-1' (Regional cluster, Aurora MySQL, us-east-1, 2 instances), 'database-1-instance-1' (Reader instance, Aurora MySQL, us-east-1b, db.r6g.2xlarge), and 'database-1-instance-1-us-east-1a' (Reader instance, Aurora MySQL, us-east-1a, db.r6g.2xlarge). A note about certificate authority end-of-life is present, along with a recommendation for Blue/Green deployments. The 'Create database' button is visible at the top right of the database list.

- Now for APP Tier
- Go to EC2
- Create an Instance
 - Launch Instance
 - ✓ Create Instance name
 - ✓ Select without keypair
 - ✓ Edit VPC and select our custom VPC
 - ✓ In subnets, Select Private APP subnet AZ1
 - ✓ Select Private Instance SG in Security group
 - ✓ Select The IAM role we created
 - ✓ Then Launch.



- Now connect to the terminal
 - Use Command `sudo -su ec2-user`
 - And Ping 8.8.8.8 to check the outside internet connectivity and google DNS servers.

```

sh-5.2$ sudo -u ec2-user
[ec2-user@ip-10-0-2-71 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=111 time=2.22 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=1.32 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=111 time=1.36 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=111 time=1.38 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=111 time=1.40 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=111 time=1.43 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=111 time=1.33 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=111 time=1.34 ms
^C
--- 8.8.8.8 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 801ms
rtt min/avg/max/mdev = 1.316/1.462/2.215/0.268 ms
[ec2-user@ip-10-0-2-71 bin]$

```

- Press Ctrl + c to stop.
- Now download the Sql package
 - ✓ Use command: `sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm`
 - ✓ Then Use command: `sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022`
 - ✓ Then Install SQL packages using this command: `sudo yum install https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm`

```

sh-5.2$ sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
--2023-09-12 19:08:08-- https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 23.0.26.132, 2600:1408:c400:1297::2e31, 2600:1408:c400:1280::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.0.26.132|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2023-09-12 19:08:09 ERROR 404: Not Found.

[ec2-user@ip-10-0-2-71 bin]$ sudo yum install mysql
Last metadata expiration check: 0:03:51 ago on Tue Sep 12 19:05:36 2023.
No match for argument: mysql
Error: Unable to find a match: mysql
[ec2-user@ip-10-0-2-71 bin]$ sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
--2023-09-12 19:12:31-- https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 23.49.176.249, 2600:1408:c400:1280::2e31, 2600:1408:c400:1297::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.49.176.249|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://repo.mysql.com/mysql57-community-release-el7-11.noarch.rpm [following]
--2023-09-12 19:12:32-- https://repo.mysql.com/mysql57-community-release-el7-11.noarch.rpm
Resolving repo.mysql.com (repo.mysql.com)... 23.206.123.16, 2600:1408:c400:1692::1d68, 2600:1408:c400:168d::1d68
Connecting to repo.mysql.com (repo.mysql.com)|23.206.123.16|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25680 (25K) [application/x-redhat-package-manager]
Saving to: 'mysql57-community-release-el7-11.noarch.rpm'

mysql57-community-release-el7-11.noar 100%[=====] 25.08K ---.KB/s in 0.007s

2023-09-12 19:12:32 (3.60 MB/s) - 'mysql57-community-release-el7-11.noarch.rpm' saved [25680/25680]

[ec2-user@ip-10-0-2-71 bin]$

```

- ✓ Then Use command `sudo yum install mysql` to install sql

```

Total download size: 35 M
Installed size: 135 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): mysql-community-common-5.7.43-1.el7.x86_64.rpm           3.1 MB/s | 313 kB   00:00
(2/4): ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86_64.rpm 2.7 MB/s | 322 kB   00:00
(3/4): mysql-community-client-5.7.43-1.el7.x86_64.rpm          14 MB/s | 2.9 MB   00:00
(4/4): mysql-community-libs-5.7.43-1.el7.x86_64.rpm            57 MB/s | 31 MB    00:00
-----
Total                                         57 MB/s | 35 MB   00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : mysql-community-common-5.7.43-1.el7.x86_64        1/4
Installing : mysql-community-libs-5.7.43-1.el7.x86_64          2/4
Running scriptlet: mysql-community-libs-5.7.43-1.el7.x86_64  2/4
Installing : ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86_64 3/4
Installing : mysql-community-client-5.7.43-1.el7.x86_64       4/4
Running scriptlet: mysql-community-client-5.7.43-1.el7.x86_64 4/4
Verifying  : ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86_64 1/4
Verifying  : mysql-community-client-5.7.43-1.el7.x86_64        2/4
Verifying  : mysql-community-common-5.7.43-1.el7.x86_64       3/4
Verifying  : mysql-community-libs-5.7.43-1.el7.x86_64         4/4

Installed:
  mysql-community-client-5.7.43-1.el7.x86_64      mysql-community-common-5.7.43-1.el7.x86_64      mysql-community-libs-5.7.43-1.el7.x86_64

Complete!
[ec2-user@ip-10-0-2-71 bin]$
```

```

Total size: 25 k
Installed size: 31 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : mysql57-community-release-el7-11.noarch        1/1
Verifying   : mysql57-community-release-el7-11.noarch        1/1

Installed:
  mysql57-community-release-el7-11.noarch

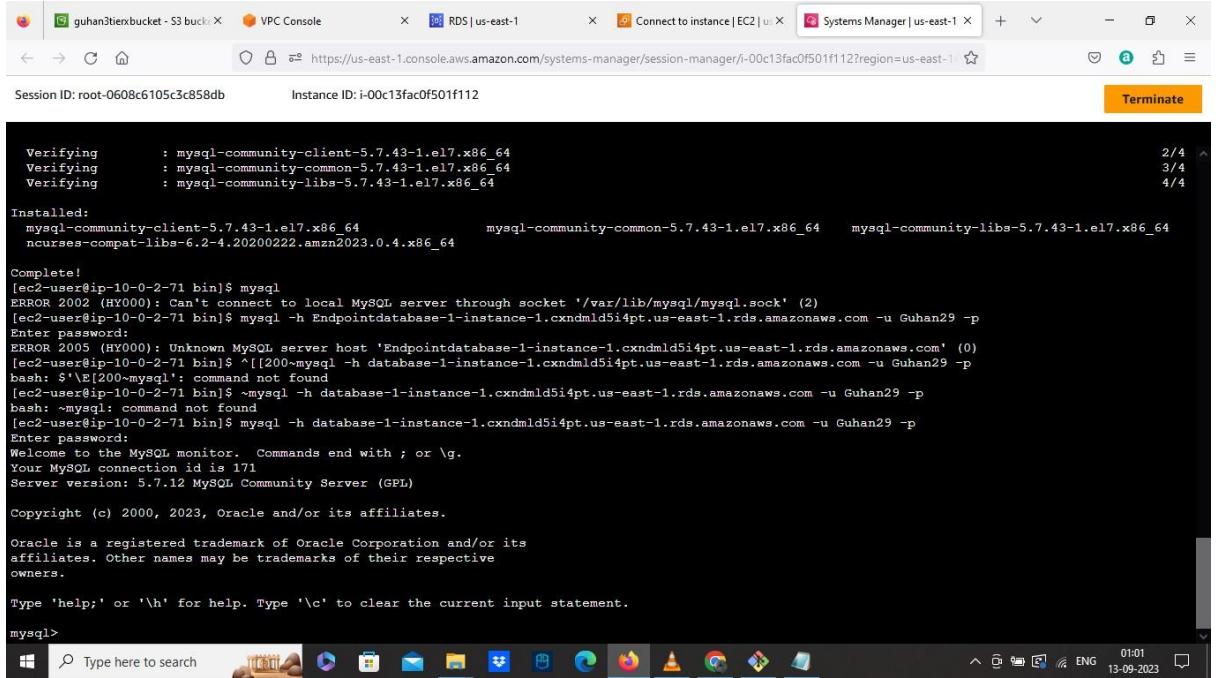
Complete!
[ec2-user@ip-10-0-2-71 bin]$ sudo yum install mysql
MySQL Connectors Community
MySQL Tools Community
MySQL 5.7 Community Server
Dependencies resolved.

=====
Package          Architecture Version       Repository      Size
=====
Installing:
  mysql-community-client      x86_64     5.7.43-1.el7      mysql57-community 31 M
Installing dependencies:
  mysql-community-common      x86_64     5.7.43-1.el7      mysql57-community 313 k
  mysql-community-libs        x86_64     5.7.43-1.el7      mysql57-community 2.9 M
  ncurses-compat-libs        x86_64     6.2-4.20200222.amzn2023.0.4      amazonlinux      322 k

Transaction Summary
=====
1 package installed:
  mysql-community-client      31 M
1 dependency installed:
  ncurses-compat-libs        322 k
1 file downloaded:
  mysql57-community-release-el7-11.noarch 313 k
```

- Use Command mysql
- Then Initiate your DB connection with your Aurora RDS writer endpoint. By the command: mysql -h database-1-instance-1.cxndmld5i4pt.us-east-1.rds.amazonaws.com -u Guhan29 -p
- Which in this command our rds writer endpoint and user name is added.
- Then Enter the Password You Already created for the database.

- Now Create a database by using this command and name it : CREATE DATABASE webappdb;



```

Verifying      : mysql-community-client-5.7.43-1.el7.x86_64
Verifying      : mysql-community-common-5.7.43-1.el7.x86_64
Verifying      : mysql-community-libs-5.7.43-1.el7.x86_64

Installed:
  mysql-community-client-5.7.43-1.el7.x86_64           mysql-community-common-5.7.43-1.el7.x86_64   mysql-community-libs-5.7.43-1.el7.x86_64
  ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86_64

Complete!
[ec2-user@ip-10-0-2-71 bin]$ mysql
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2)
[ec2-user@ip-10-0-2-71 bin]$ mysql -h Endpointdatabase-1-instance-1.cxndmld5i4pt.us-east-1.rds.amazonaws.com -u Guhan29 -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'Endpointdatabase-1-instance-1.cxndmld5i4pt.us-east-1.rds.amazonaws.com' (0)
[ec2-user@ip-10-0-2-71 bin]$ ^C
[ec2-user@ip-10-0-2-71 bin]$ ~mysql -h database-1-instance-1.cxndmld5i4pt.us-east-1.rds.amazonaws.com -u Guhan29 -p
bash: ~mysql: command not found
[ec2-user@ip-10-0-2-71 bin]$ ~mysql -h database-1-instance-1.cxndmld5i4pt.us-east-1.rds.amazonaws.com -u Guhan29 -p
bash: ~mysql: command not found
[ec2-user@ip-10-0-2-71 bin]$ mysql -h database-1-instance-1.cxndmld5i4pt.us-east-1.rds.amazonaws.com -u Guhan29 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 171
Server version: 5.7.12 MySQL Community Server (GPL)

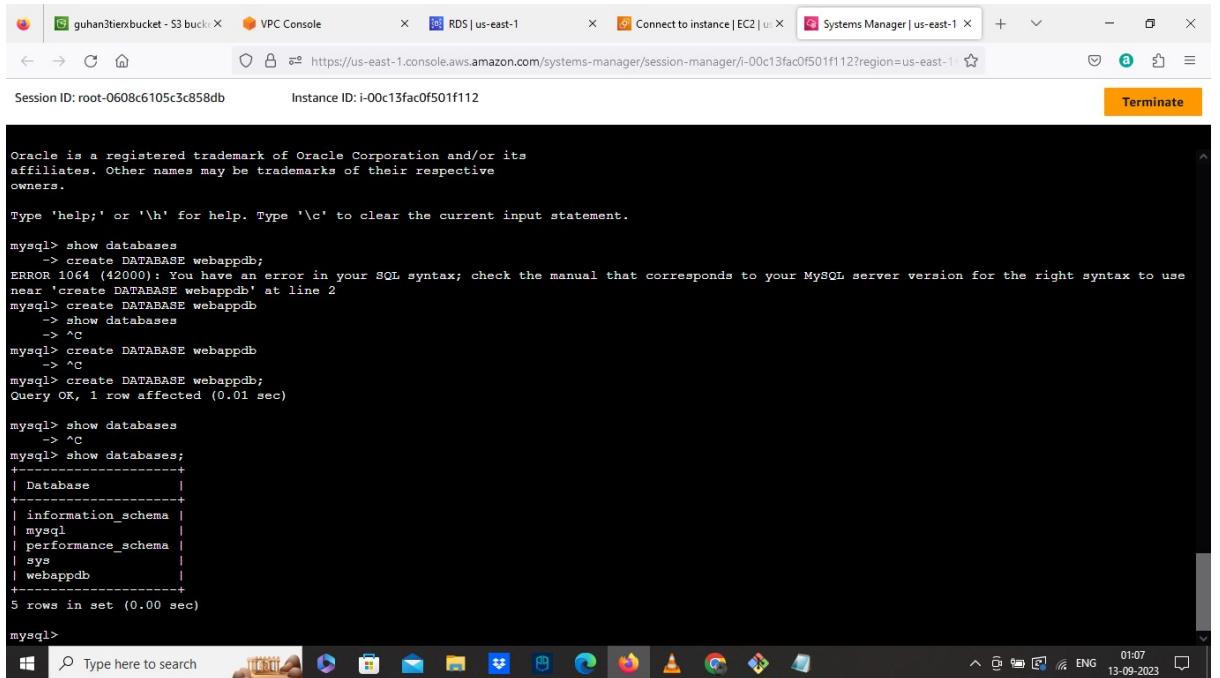
Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>

```



```

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases
-> create DATABASE webappdb;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use
near 'create DATABASE webappdb' at line 2
mysql> create DATABASE webappdb
-> show databases
-> ^C
mysql> create DATABASE webappdb;
-> ^C
mysql> create DATABASE webappdb;
Query OK, 1 row affected (0.01 sec)

mysql> show databases
-> ^C
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| webappdb |
+-----+
5 rows in set (0.00 sec)

mysql>

```

- Now Use this command to check whether the database we created is available: SHOW DATABASES;
- Change to your database by using this command: USE webappdb;

```

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

mysql> use webappdb
Database changed
mysql> 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 (0.03 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_webappdb |
+-----+
| transactions      |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO transactions (amount,description) VALUES ('400','groceries');
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM transactions;
+-----+
| id | amount | description |
+-----+
| 1  | 400.00 | groceries   |
+-----+
1 row in set (0.00 sec)

mysql> 

```

- Create a table by using this command: `CREATE TABLE IF NOT EXISTS transactions(id INT NOT NULL AUTO_INCREMENT, amount DECIMAL(10,2), description VARCHAR(100), PRIMARY KEY(id));`
- Check the created table: `SHOW TABLES;`
- Insert the data in the table by using this command: `INSERT INTO transactions (amount,description) VALUES ('400','groceries');`
- Check The data added by: `SELECT * FROM transactions;`
- Then type exit to exit the SQL.

```

+-----+
5 rows in set (0.00 sec)

mysql> use webappdb
Database changed
mysql> 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 (0.03 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_webappdb |
+-----+
| transactions      |
+-----+
1 row in set (0.00 sec)

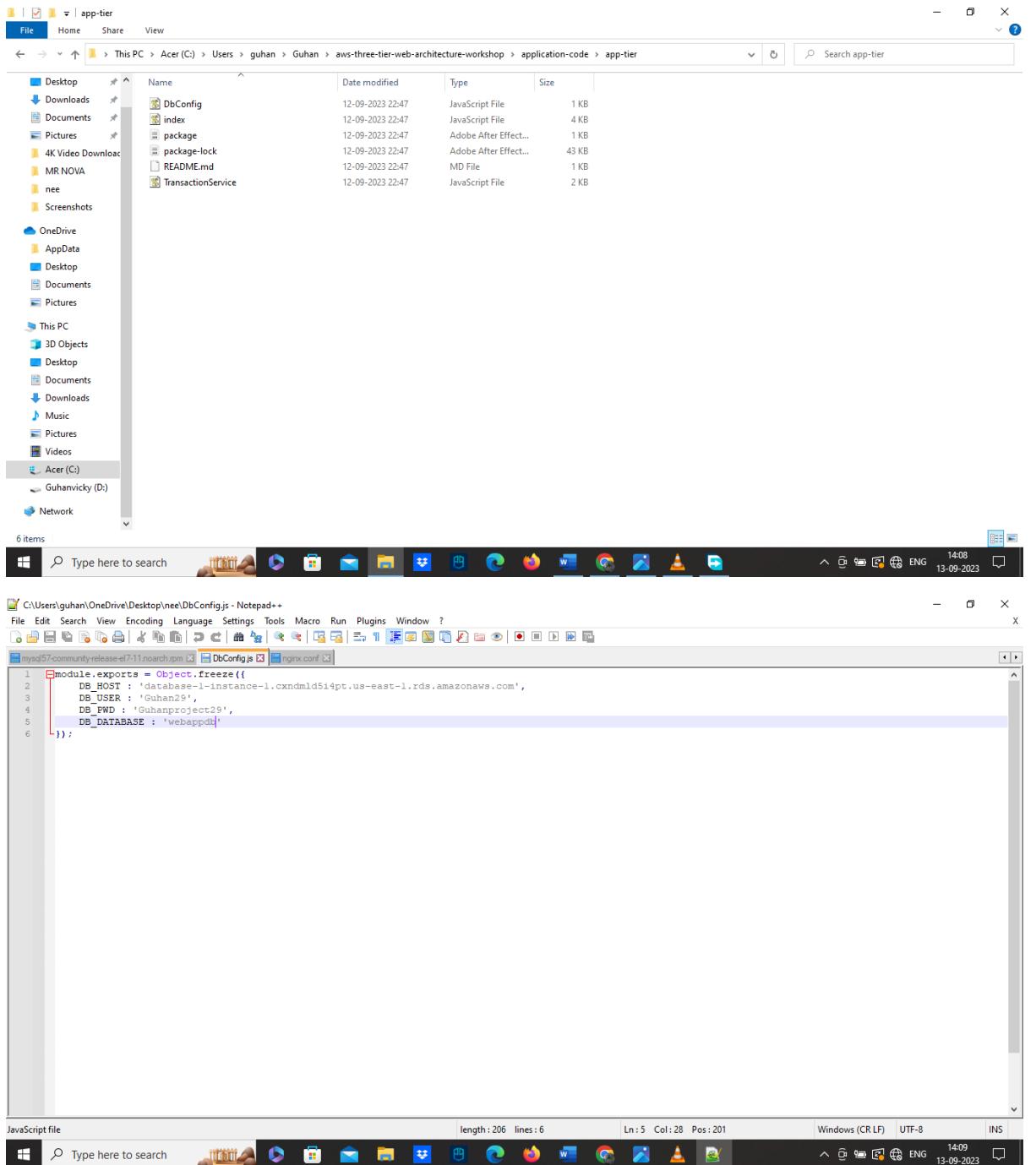
mysql> INSERT INTO transactions (amount,description) VALUES ('400','groceries');
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM transactions;
+-----+
| id | amount | description |
+-----+
| 1  | 400.00 | groceries   |
+-----+
1 row in set (0.00 sec)

mysql> clear
mysql> ^C
mysql> clear
mysql> exit

```

- Then go to S3 bucket we created
- The DB config file which we already downloaded from GitHub, need to edit before uploading in S3.
- Where the details will be empty which we downloaded from Github
- Now open the DB config file with Notepad++
- Then Add the required details of the Database we created, add Writer endpoint in the DB host Field and webappdb in DB database field.
- After that save the file, and upload to S3 bucket we created.



- Now Upload the app tier folder to the S3 bucket also.

Files and folders (6 Total, 48.8 KB)

Name	Folder	Type	Size	Status	Error
index.js	app-tier/	application/x-javascript	3.2 KB	Succeeded	-
package-lock.json	app-tier/	application/json	42.9 KB	Succeeded	-
package.json	app-tier/	application/json	682.0 B	Succeeded	-
README.md	app-tier/	-	14.0 B	Succeeded	-
TransactionService.js	app-tier/	application/x-javascript	1.8 KB	Succeeded	-
DbConfig.js	-	application/x-javascript	206.0 B	Succeeded	-

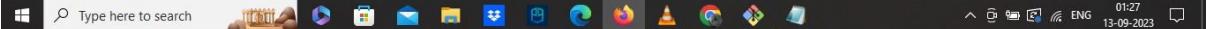


- Also move the config file to app tier folder
- Now we need to install necessary components to run in backend
- Install Node version manager by running this command: curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
- Then we need to instal Node.js and use it by running this command: nvm install 16
nvm use 16
- Install pm2, which make the nodejs app active even when we exit the instance. Use this command to install: npm install -g pm2

```
[ec2-user@ip-10-0-2-71 bin]$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total   Spent    Left Speed
100 14926  100 14926    0     0  303k      0 --:--:-- --:--:-- 310k
=> 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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-2-71 bin]$
```



```
[ec2-user@ip-10-0-2-71 bin]$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total   Spent    Left  Speed
100 14926  100 14926    0     0  303k  0:--:-- --:--:--:--:--:-- 310k
=> 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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-2-71 bin]$ nvm install 16
nvm use 16
Downloading and installing node v16.20.2...
Downloading 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)
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-2-71 bin]$
```

```
=> 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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-2-71 bin]$ nvm install 16
nvm use 16
Downloading and installing node v16.20.2...
Downloading 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)
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-2-71 bin]$ npm install -g pm2
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

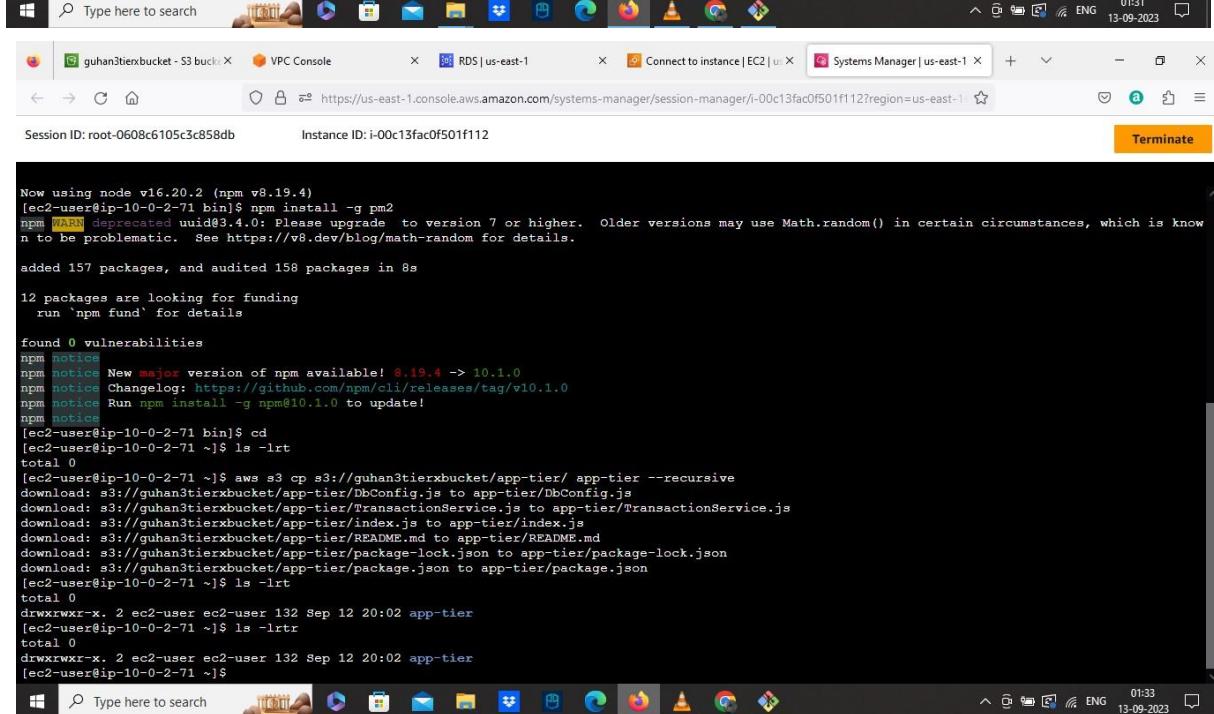
added 157 packages, and audited 158 packages in 8s

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

found 0 vulnerabilities

npm notice New major version of npm available! 8.19.4 -> 10.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0
npm notice Run npm install -g npm@10.1.0 to update!
npm notice
[ec2-user@ip-10-0-2-71 bin]$
```

- Change the directory cd
- Now we need to fetch and download the code from the S3 bucket, so we need to use this command: aws s3 cp s3://guhan3tierxbucket/app-tier/ app-tier -- recursive
- Add our bucket name in the command and run
- Now download the entire folder ls -ltr
- Then Switch to app tier by using cd app-tier/
- Activate the pm2 we installed by using this command: pm2 start index.js



Session ID: root-0608c6105c3c858db Instance ID: i-00c13fac0f501f112

Terminate

```

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-2-71 bin]$ nvm install 16
nvm use 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)
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-2-71 bin]$ npm install -g pm2
npm [WARN] deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 157 packages, and audited 158 packages in 8s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 8.19.4 -> 10.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0
npm notice Run npm install -g npm@10.1.0 to update!
npm notice
[ec2-user@ip-10-0-2-71 bin]$ cd
[ec2-user@ip-10-0-2-71 ~]$ ls -lrt
total 0
[ec2-user@ip-10-0-2-71 ~]$ aws s3 cp s3://guhan3tierxbucket/app-tier/ app-tier --recursive

```

Type here to search

Session ID: root-0608c6105c3c858db Instance ID: i-00c13fac0f501f112

Terminate

```

Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-2-71 bin]$ npm install -g pm2
npm [WARN] deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 157 packages, and audited 158 packages in 8s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 8.19.4 -> 10.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0
npm notice Run npm install -g npm@10.1.0 to update!
npm notice
[ec2-user@ip-10-0-2-71 bin]$ cd
[ec2-user@ip-10-0-2-71 ~]$ ls -lrt
total 0
[ec2-user@ip-10-0-2-71 ~]$ aws s3 cp s3://guhan3tierxbucket/app-tier/ app-tier --recursive
download: s3://guhan3tierxbucket/app-tier/DbConfig.js to app-tier/DbConfig.js
download: s3://guhan3tierxbucket/app-tier/TransactionsService.js to app-tier/TransactionsService.js
download: s3://guhan3tierxbucket/app-tier/index.js to app-tier/index.js
download: s3://guhan3tierxbucket/app-tier/README.md to app-tier/README.md
download: s3://guhan3tierxbucket/app-tier/package-lock.json to app-tier/package-lock.json
download: s3://guhan3tierxbucket/app-tier/package.json to app-tier/package.json
[ec2-user@ip-10-0-2-71 ~]$ ls -lrt
total 0
drwxrwxr-x. 2 ec2-user ec2-user 132 Sep 12 20:02 app-tier
[ec2-user@ip-10-0-2-71 ~]$ ls -lrt
total 0
drwxrwxr-x. 2 ec2-user ec2-user 132 Sep 12 20:02 app-tier
[ec2-user@ip-10-0-2-71 ~]$
```

Type here to search

```
[ec2-user@ip-10-0-2-71 ~]$ cd /app-tier
bash: cd: /app-tier: No such file or directory
[ec2-user@ip-10-0-2-71 ~]$ cd app-tier/
[ec2-user@ip-10-0-2-71 app-tier]$ pm2 start index.js

-----
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:
```

- To check the app is running, run the command: **pm2 list** and check the online and error status by using this command: **pm2 logs**
- Now make the app to keep running even after exit, run this command: **pm2 startup** and run the command it display next.

```
PM2 | 2023-09-12T20:04:47: PM2 log: App [index:0] starting in -fork mode-
PM2 | 2023-09-12T20:04:47: PM2 log: App [index:0] online
PM2 | 2023-09-12T20:04:47: PM2 log: App [index:0] exited with code [1] via signal [SIGINT]
PM2 | 2023-09-12T20:04:47: PM2 log: App [index:0] starting in -fork mode-
PM2 | 2023-09-12T20:04:47: PM2 log: App [index:0] online
PM2 | 2023-09-12T20:04:48: PM2 log: App [index:0] exited with code [1] via signal [SIGINT]
PM2 | 2023-09-12T20:04:48: PM2 log: Script /home/ec2-user/app-tier/index.js had too many unstable restarts (16). Stopped. "errored"
PM2 | 2023-09-12T20:07:42: PM2 log: App [index:0] starting in -fork mode-
PM2 | 2023-09-12T20:07:42: PM2 log: App [index:0] online

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:
0|index |   at Module._requireModule.require (/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/node_modules/require-in-the-middle/index.js:81:25)
0|index |   at require (node:internal/modules/cjs/helpers:119:18)
0|index |   at Object.<anonymous> (/home/ec2-user/app-tier/TransactionService.js:2:15)
0|index |   at Module._compile (node:internal/modules/cjs/loader:1198:14)
0|index |   at Object.Module._extensions..js (node:internal/modules/cjs/loader:1252:10)
0|index |   at Module.load (node:internal/modules/cjs/loader:1076:32)
0|index |   at Function.Module._load (node:internal/modules/cjs/loader:911:12)
0|index |   at Module.require (node:internal/modules/cjs/loader:1100:19)
0|index |   at Module._requireModule.require (/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/node_modules/require-in-the-middle/index.js:101:39)
0|index |   code: 'MODULE_NOT_FOUND',
0|index |   requireStack: [
0|index |     '/home/ec2-user/app-tier/TransactionService.js',
0|index |     '/home/ec2-user/app-tier/index.js'
0|index |   ]
0|index | }

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

```

Session ID: root-0608c6105c3c858db
Instance ID: i-00c13fac0f501f112
Terminate

PM2    | 2023-09-12T20:04:48: PM2 log: Script /home/ec2-user/app-tier/index.js had too many unstable restarts (16). Stopped. "errored"
PM2    | 2023-09-12T20:07:42: PM2 log: App [index:0] starting in -fork mode-
PM2    | 2023-09-12T20:07:42: PM2 log: App [index:0] online

/home/ec2-user/.pm2/logs/index-error.log last 15 lines:
0|index  | at Module.Hook._requireModule.require (/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/node_modules/require-in-the-middle/index.js:81:25)
0|index  |   at require (node:internal/modules/cjs/helpers:119:18)
0|index  |   at Object.<anonymous> (/home/ec2-user/app-tier/transactionService.js:2:15)
0|index  |   at Module._compile (node:internal/modules/cjs/loader:1198:14)
0|index  |   at Object.Module._extensions..js (node:internal/modules/cjs/loader:1252:10)
0|index  |   at Module.load (node:internal/modules/cjs/loader:1076:32)
0|index  |   at Function.Module._load (node:internal/modules/cjs/loader:911:12)
0|index  |   at Module.require (node:internal/modules/cjs/loader:1100:19)
0|index  |   at Module.Hook._requireModule.require (/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/node_modules/require-in-the-middle/index.js:101:39) {
0|index  |     code: 'MODULE_NOT_FOUND',
0|index  |     requireStack: [
0|index  |       '/home/ec2-user/app-tier/TransactionService.js',
0|index  |       '/home/ec2-user/app-tier/index.js'
0|index  |     ]
0|index  |   }

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

^C
[ec2-user@ip-10-0-2-71 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup sys
temd -u ec2-user -hp /home/ec2-user
[ec2-user@ip-10-0-2-71 app-tier]$ 

```

- Save the configuration by: **pm2 save**
- Test the app health by : **curl <http://localhost:4000/health>**
- Test the database connection by using: **curl <http://localhost:4000/transaction>**
- Then the app layer will be created.

```

Session ID: root-0608c6105c3c858db
Instance ID: i-00c13fac0f501f112
Terminate

:/usr/bin:/sbin:/bin:/var/lib/snapd/snap/bin:/home/ec2-user/.nvm/versions/node/v16.20.2/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/home/ec2-user/.pm2
PIDFile=/home/ec2-user/.pm2/pm2.pid
Restart=on-failure

ExecStart=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-ec2-user.service
Command list
[ 'systemctl enable pm2-ec2-user' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-ec2-user.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-ec2-user...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-ec2-user.service → /etc/systemd/system/pm2-ec2-user.service.
[PM2] [v] Command successfully executed.
+-----+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
[ec2-user@ip-10-0-2-71 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[ec2-user@ip-10-0-2-71 app-tier]$ curl http://localhost:4000/health
"This is the health check"[ec2-user@ip-10-0-2-71 app-tier]$ curl http://localhost:4000/transaction
{"result":[{"id":1,"amount":400,"description":"groceries"}]}[ec2-user@ip-10-0-2-71 app-tier]$ 

```

- Go to EC2
 - ➔ Select the Instance we created and create an Image AMI

Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	3000	0	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

- Go to Target Groups
 - ➔ Choose Target type as Instance.
 - ➔ Give a name to the target group we created
 - ➔ Set the port to 4000 and protocol to HTTP
 - ➔ Select our custom VPC
 - ➔ Change the values to 2 in healthy threshold and the TG will be created.

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

Instances

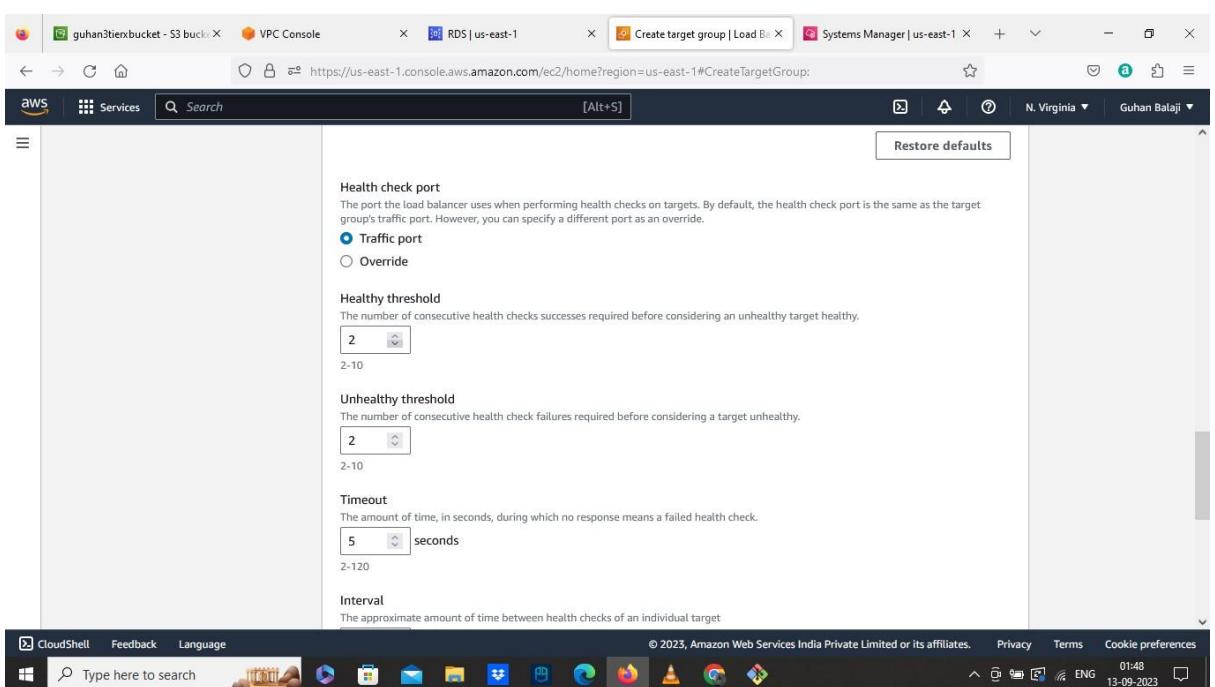
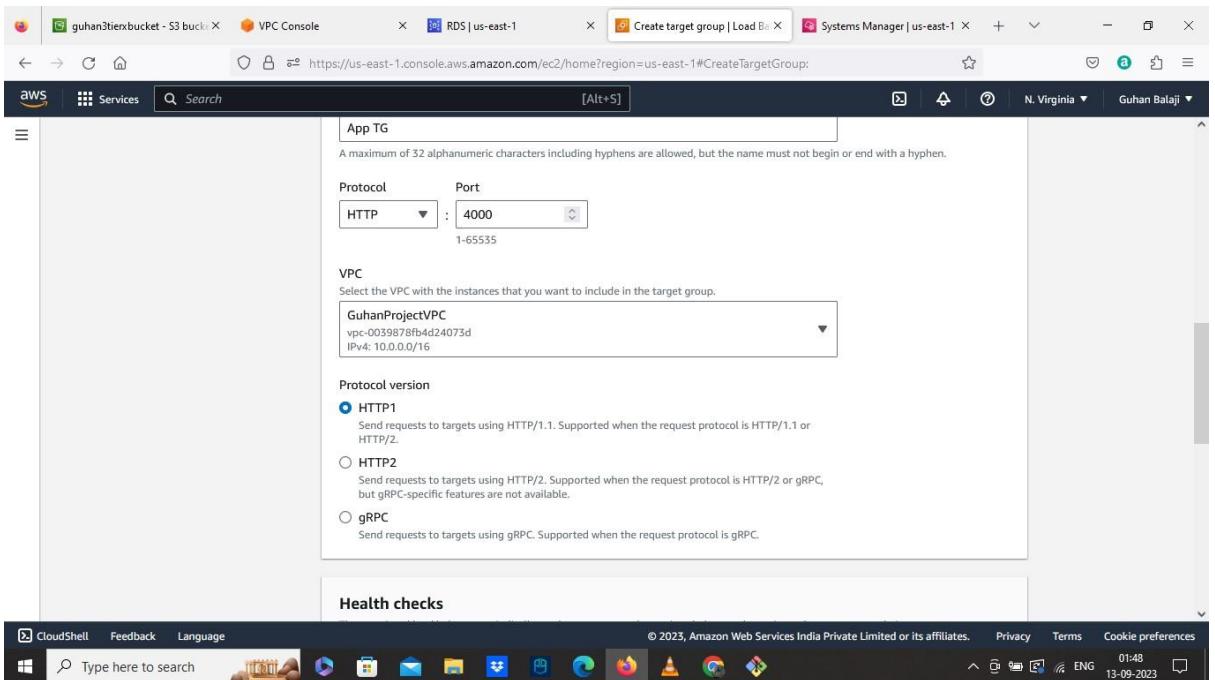
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

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, simplifying inter-application communication.
- 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.



- **Create Load Balancer**
 - ➔ Select the scheme as Internal
 - ➔ Select our custom VPC
 - ➔ In Mapping, Select the two private App Subnet az1, az2
 - ➔ Select the existing Internal Load Balancer Security Group
 - ➔ In Listeners and routing, Set the port to 80 and select The Target group we created in Forward.

Screenshot of the AWS CloudShell interface showing the configuration of an Application Load Balancer (ALB) in the N. Virginia region. The user is on the 'Mappings' step of the wizard.

Mappings

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az4)

Subnet: subnet-0d3cc9fa0ef9e5175 Private App Subnet AZ-1

IPv4 address: Assigned from CIDR 10.0.2.0/24

us-east-1b (use1-az6)

Subnet: subnet-05375332dd95c0d3c Private App Subnet AZ-2

IPv4 address: Assigned from CIDR 10.0.3.0/24

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 01:51 13-09-2023

Screenshot of the AWS CloudShell interface showing the continuation of ALB configuration.

Security groups

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups: Select up to 5 security groups

Internal Load balancer SG
sg-034231f116aa3befb VPC: vpc-0039878fb4d24073d

Listeners and routing

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 01:52 13-09-2023

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there are tabs for 'Metrics' and 'Logs'. Below the tabs, a search bar contains the query 'CloudWatch Metrics'. The main area displays a table of metrics with columns for 'Metric Name', 'Dimensions', 'Unit', and 'Period'. One row is selected, showing detailed information about the metric 'AWS/CloudWatchMetrics/CloudWatchMetricsLogEventCount'. The 'CloudWatch Metrics Log Events' section shows log events for this metric, with one event highlighted. The bottom of the screen shows the AWS navigation bar and the status bar indicating the date and time.

The screenshot shows the AWS Lambda function configuration page. The left sidebar lists functions: 'Create New Function', 'Lambda@Edge', 'lambda-test', and 'lambda-test-1'. The main content area shows the configuration for the 'lambda-test' function. It includes sections for 'Basic Settings', 'Code', 'Environment', 'Triggers', and 'Permissions'. The 'Basic Settings' section shows the function name 'lambda-test', runtime 'Node.js 14.x', and memory '128 MB'. The 'Code' section shows the ARN 'arn:aws:lambda:us-east-1:123456789012:function:lambda-test'. The 'Environment' section shows environment variables like 'AWS_LAMBDA_FUNCTION_NAME' and 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE'. The 'Triggers' section shows a trigger for 'lambda-test-1'. The 'Permissions' section shows the policy 'lambda-test-1'. The bottom of the screen shows the AWS navigation bar and the status bar indicating the date and time.

- Create The Launch Template
 - ➔ Select our EMI from App Instance
 - ➔ Set Instance Type as Free Tier
 - ➔ Select The Private Instance SG in Security Group
 - ➔ Choose The IAM role we created

guhan3tierbucket - S3 bucket X VPC Console X RDS | us-east-1 X Create launch template | EC2 X Systems Manager | us-east-1 X

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

aws Services Search [Alt+S]

Recent My AMIs Quick Start

Don't include in launch template Owned by me Shared with me

Amazon Machine Image (AMI)

AWSappimage
ami-0efd4af5d811b3af5
2023-09-12T20:16:55.000Z Virtualization: hvm ENA enabled: true Root device type: ebs boot mode: uefi-preferred

Description
AWS 3 tier image

Architecture x86_64 AMI ID ami-0efd4af5d811b3af5

Summary

Software Image (AMI)
AWS 3 tier image
ami-0efd4af5d811b3af5

Virtual server type (instance type)
-

Firewall (security group)
-

Storage (volumes)
1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per

Create launch template

guhan3tierbucket - S3 bucket X VPC Console X RDS | us-east-1 X Create launch template | EC2 X Systems Manager | us-east-1 X

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

aws Services Search [Alt+S]

Don't include in launch template Create new subnet ↗

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group Create security group

Security groups Info

Select security groups

Private SG sg-09cb108e44cf349d2 X

VPC: vpc-0039878fb4d24073d

Advanced network configuration

▼ Storage (volumes) Info

EBS Volumes Hide details

Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

Summary

Software Image (AMI)
AWS 3 tier image
ami-0efd4af5d811b3af5

Virtual server type (instance type)
t2.micro

Firewall (security group)
Private SG

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per

Create launch template

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#FunctionConfiguration:>. The page displays the configuration for a function named 'app-Guhan'. The 'Handler' dropdown is set to 'lambda.lambda_handler'. The 'Memory Size' is 128 MB, and the 'Timeout' is 10 seconds. The 'Code' section shows the ARN 'arn:aws:lambda:us-east-1:0831...:function:app-Guhan'. The 'Environment' section lists environment variables: 'AWS_LAMBDA_FUNCTION_NAME' (value: app-Guhan), 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE' (value: 128), 'AWS_LAMBDA_FUNCTION_TIMEOUT' (value: 10), and 'AWS_LAMBDA_FUNCTION_SOURCE' (value: /tmp/app-Guhan.zip). The 'Logs' tab is selected, showing log entries for the function's execution.

- Create AutoScaling groups
 - ➔ Select the Launch template we created
 - ➔ Select our VPC
 - ➔ Select the two private subnets az1,az2
 - ➔ Select the existing load balancer we created
 - ➔ Select the target groups we created
 - ➔ Set values as 2 in Group Size
 - ➔ Then the ASG will be created.

The screenshot shows the AWS Auto Scaling console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:>. The page is titled 'Step 3: Configure advanced options'. It displays the following configuration:

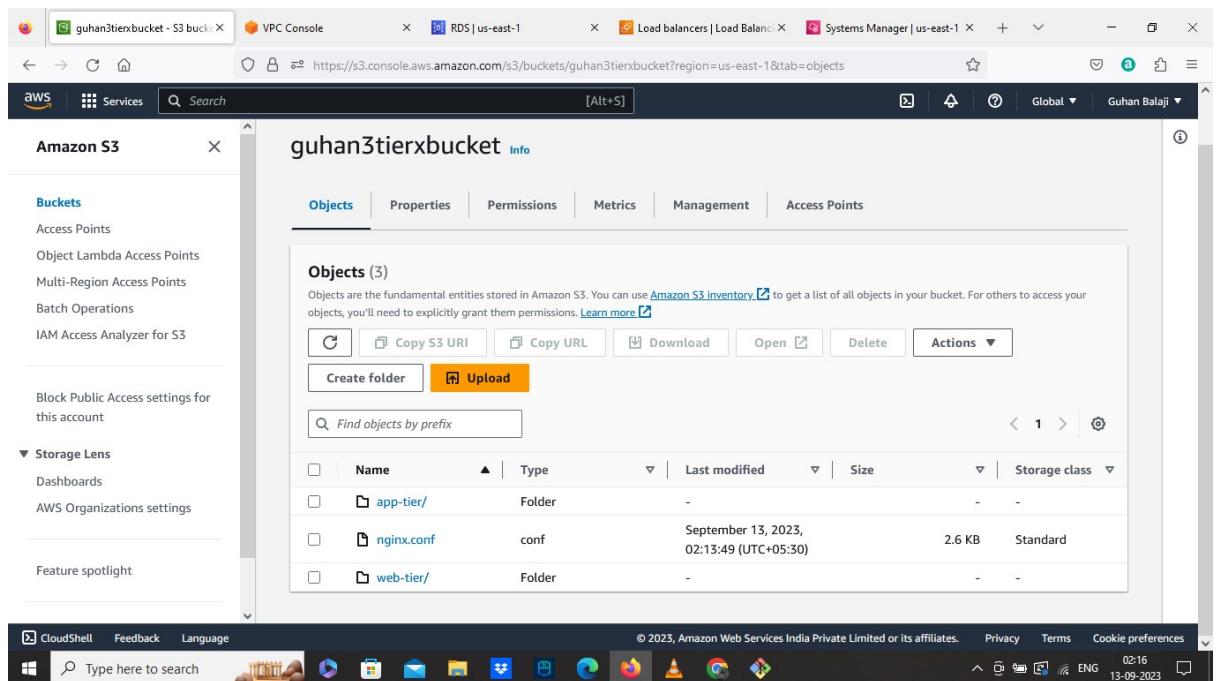
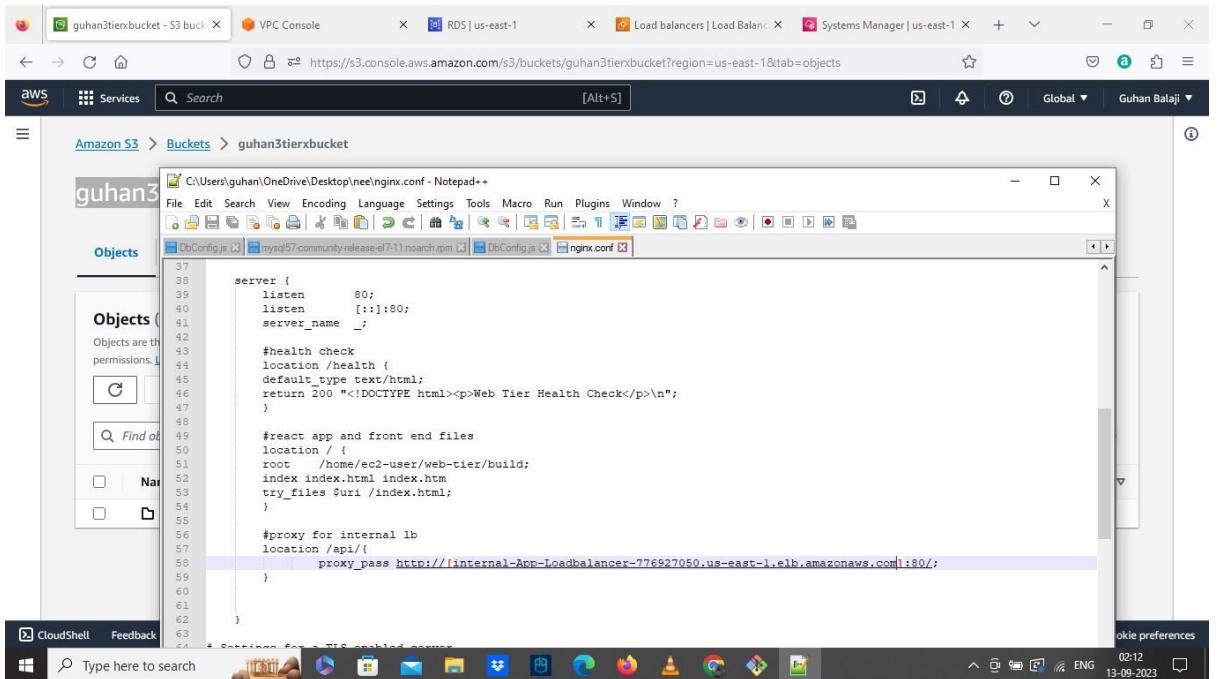
- Load balancing**:
 - Load balancer 1: Name - App-Loadbalancer, Type - Application/HTTP, Target group - App-TG
- VPC Lattice integration options**:
 - VPC Lattice target groups: -
- Health checks**:
 -

The status bar at the bottom indicates the time is 02:00 on 13-09-2023.

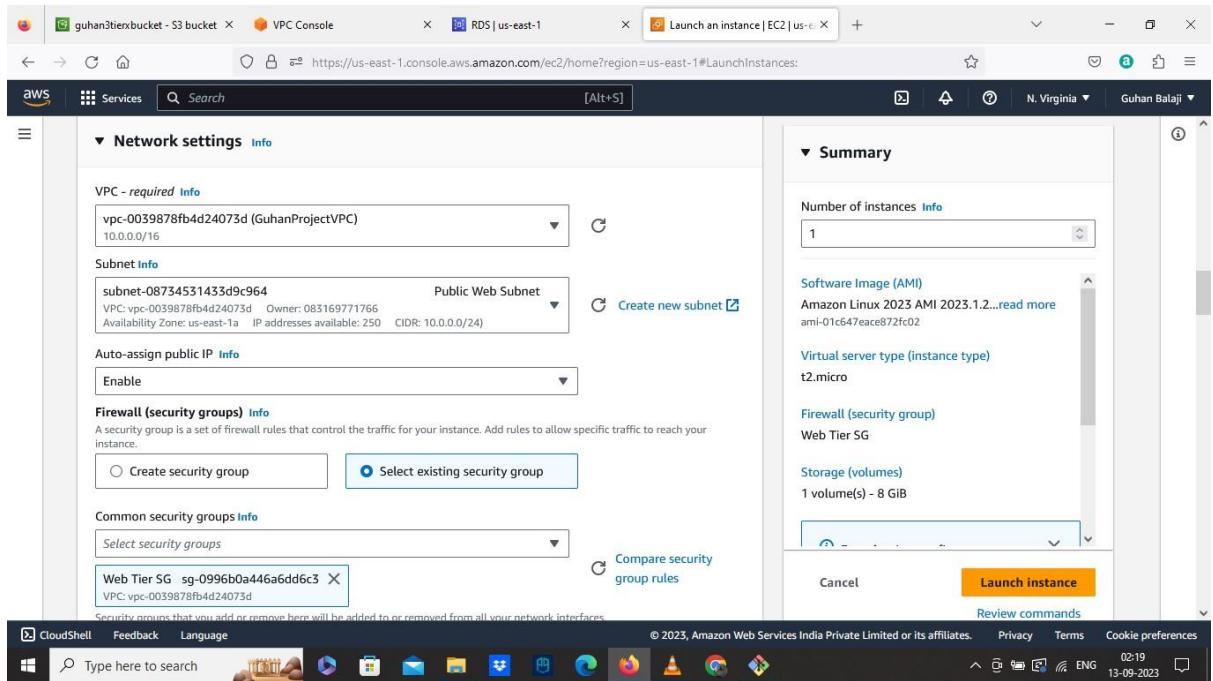
The screenshot shows the AWS Auto Scaling Groups page. At the top, there are tabs for VPC Console, RDS | us-east-1, Auto Scaling groups | EC2, Systems Manager | us-east-1, and others. The main content area has a header "Auto Scaling groups (1) Info". Below it is a search bar and a table with columns: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Last update. One row is visible for "App-GuhanASG" with a status of "app-Guhan | Version Default". A message at the bottom says "0 Auto Scaling groups selected".

The screenshot shows the AWS Instances page. At the top, there are tabs for VPC Console, RDS | us-east-1, Instances | EC2 | us-east-1, Systems Manager | us-east-1, and others. A green banner at the top states "Currently creating AMI ami-0efd4af5d811b3af5 from instance i-00c13fac0f501f112. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI." The main content area has a header "Instances (3) Info". Below it is a search bar and a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Three instances are listed: "i-08884cc8b3af2b66e" (Running, t2.micro, Initializing, No alarms, us-east-1a), "i-0e688eb86ff4f3a6b" (Running, t2.micro, Initializing, No alarms, us-east-1b), and "i-00c13fac0f501f112" (Running, t2.micro, 2/2 checks passed, No alarms, us-east-1a). A modal window titled "Select an instance" is open in the foreground.

- Now Update the nginx.conf file where we downloaded.
 - Open the nginx.conf file with notepad++ or can also be edited with gitbash, here iam using notepad++
 - Scroll down and find the Replace ILB DNS with your Internal Load balancer DNS and save it
 - The Modified nginx.conf file is uploaded to our S3 bucket
 - Also Upload the WEB tier folder to our S3 bucket.



- Now create the WEB instance
 - ➔ Launch an Instance
 - ➔ Select our VPC
 - ➔ Here select the Public web subnet az1 in subnet
 - ➔ Enable the Auto assign the public IP
 - ➔ Select the existing security group which is WEB tier SG
 - ➔ Select our IAM role.



- Connect to the terminal
 - ➔ Run the command `sudo -su ec2-user`
 - ➔ Ping 8.8.8.8 to check the outside internet connectivity and google DNS servers
 - ➔ Now we need to install the components for front end, again install NVM by running these commands: `curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash`
`source ~/.bashrc`
`nvm install 16`
`nvm use 16`
 - ➔ Change the directory `cd`
 - ➔ Download the codes from our S3 Bucket web tier folder by running this command: `aws s3 cp s3://guhan3tierxbucket/web-tier/ web-tier – recursive`
 - ➔ Create Build folder in Web folder for react app by running this commands: `cd web-tier/`
`npm install`
`npm run build`

```
sh-5.2$ sudo -su ec2-user
[ec2-user@ip-10-0-0-27 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=55 time=1.60 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=1.54 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=1.56 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=1.63 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=55 time=1.65 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.538/1.594/1.649/0.041 ms
[ec2-user@ip-10-0-0-27 bin]$
```

```
sh-5.2$ sudo -su ec2-user
[ec2-user@ip-10-0-0-27 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=55 time=1.60 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=1.54 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=1.56 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=1.63 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=55 time=1.65 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.538/1.594/1.649/0.041 ms
[ec2-user@ip-10-0-0-27 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 253k 0 --:--:--:--:--:-- 255k
=> 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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-0-27 bin]$
```

```
[ec2-user@ip-10-0-0-27 bin]$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
       0      0      0      0      0      0      0      0      0      0      0      0
  100 14926  100 14926    0      0  253k    0 --:--:--:--:--:--:--:-- 255k
=> 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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-0-27 bin]$ source ~/.bashrc
[ec2-user@ip-10-0-0-27 bin]$ nvm install 16
nvm use 16
Downloading and installing node v16.20.2...
Downloading 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)
Now using node v16.20.2 (npm v8.19.4)
[ec2-user@ip-10-0-0-27 bin]$ cd
[ec2-user@ip-10-0-0-27 ~]$ aws s3 cp s3://guhan3tierbucket/web-tier/ web-tier --recursive
[ec2-user@ip-10-0-0-27 ~]$
```

```
this plugin is no longer maintained. Please use @babel/plugin-transform-private-methods instead.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1482 packages, and audited 1483 packages in 1m

247 packages are looking for funding
  run `npm fund` for details

6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice New major version of npm available! 8.19.4 -> 10.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0
npm notice Run npm install -g npm@10.1.0 to update!
npm notice
[ec2-user@ip-10-0-0-27 web-tier]$
```

- Install NGINX where this will help to direct our API calls to the internal load balancer.
- Run the command: `sudo yum install nginx^c`
- So now we have to replace the default nginx configuration file from the Nginx we installed with our modified configuration `nginx.conf` file, where we changed our load balancer DNS name.
- Use the command to navigate: `cd /etc/nginx` then list `ls` and remove the file from its location by running this command: `sudo rm nginx.conf`
- Download the modified from our S3 Bucket using:

```
sudo aws s3 cp s3://guhan3tierxbucket/nginx.conf .
```

```
Find out more about deployment here:
https://cra.link/deployment

[ec2-user@ip-10-0-0-27 web-tier]$ sudo yum install nginx
Last metadata expiration check: 0:12:23 ago on Tue Sep 12 20:50:24 2023.
Dependencies resolved.

=====
Package           Architecture      Version       Repository   Size
=====
Installing:
  nginx           x86_64          1:1.24.0-1.amzn2023.0.1   amazonlinux  32 k
Installing dependencies:
  generic-logos-httd noarch        18.0.0-12.amzn2023.0.3   amazonlinux  19 k
  gperftools-libs  x86_64         2.9.1-1.amzn2023.0.2   amazonlinux 309 k
  libunwind         x86_64         1.4.0-5.amzn2023.0.2   amazonlinux  66 k
  nginx-core        x86_64         1:1.24.0-1.amzn2023.0.1  amazonlinux 586 k
  nginx-filesystem noarch        1:1.24.0-1.amzn2023.0.1  amazonlinux  9.0 k
  nginx-mime types noarch        2.1.49-3.amzn2023.0.3   amazonlinux  21 k

Transaction Summary
=====
Install  7 Packages

Total download size: 1.0 M
```

```
-rw-r--r--. 1 root root 2837 May 31 21:09 koi-utf
-rw-r--r--. 1 root root 1007 May 31 21:09 fastcgi_params.default
-rw-r--r--. 1 root root 1007 May 31 21:09 fastcgi_params
-rw-r--r--. 1 root root 1077 May 31 21:09 fastcgi.conf.default
-rw-r--r--. 1 root root 1077 May 31 21:09 fastcgi.conf
drwxr-xr-x. 2 root root 6 May 31 21:09 default.d
drwxr-xr-x. 2 root root 6 May 31 21:09 conf.d
[ec2-user@ip-10-0-0-27 nginx]$ sudo nginx.conf nginx.conf_bkp
sudo: nginx.conf: command not found
[ec2-user@ip-10-0-0-27 nginx]$ sudo nginx.conf nginx.conf_bkp
sudo: nginx.conf: command not found
[ec2-user@ip-10-0-0-27 nginx]$ sudo rm nginx.conf
[ec2-user@ip-10-0-0-27 nginx]$ sudo aws s3 cp s3://guhan3tierxbucket/nginx.conf

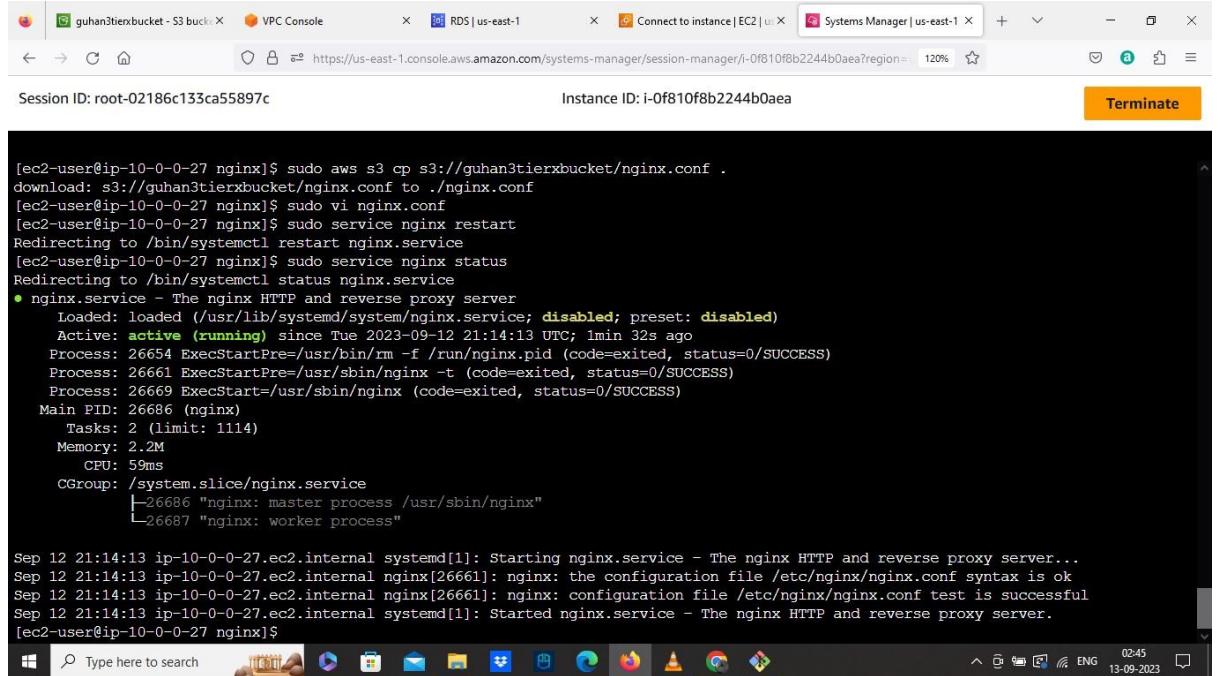
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help

aws: error: the following arguments are required: paths

[ec2-user@ip-10-0-0-27 nginx]$ sudo aws s3 cp s3://guhan3tierxbucket/nginx.conf .
download: s3://guhan3tierxbucket/nginx.conf to ./nginx.conf
[ec2-user@ip-10-0-0-27 nginx]$
```

- Restart the Nginx by running this command: `sudo service nginx restart`
- Check the status of Nginx with this command: `sudo service nginx status`

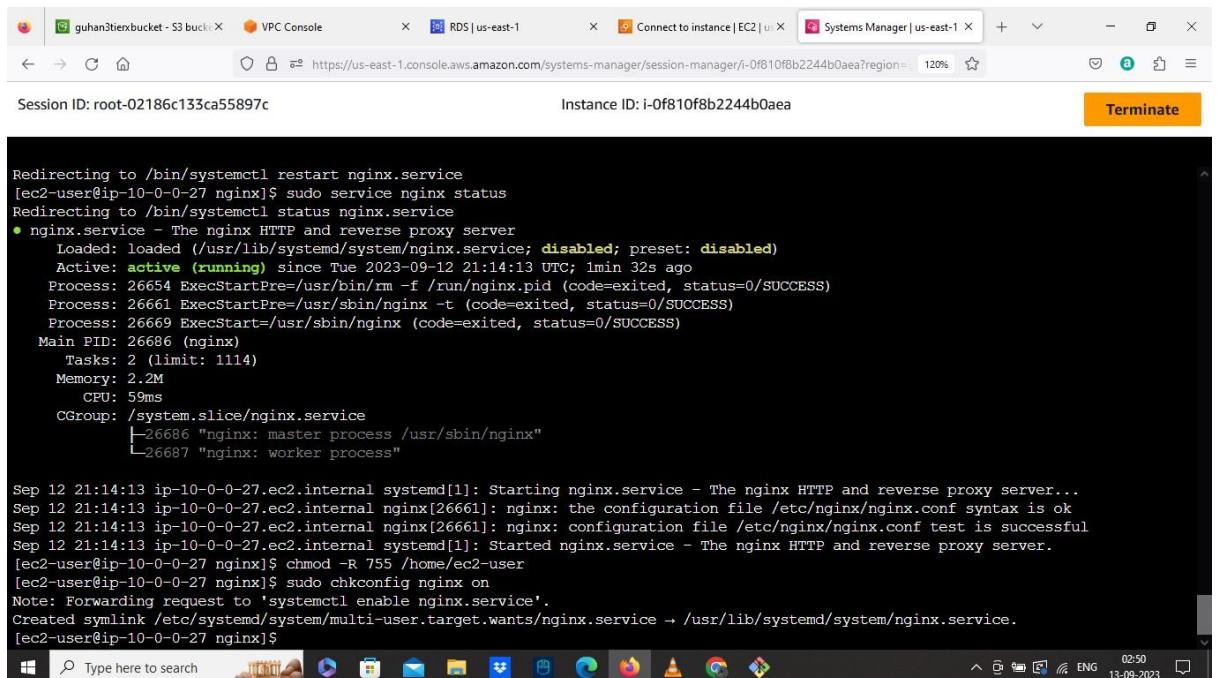


```
[ec2-user@ip-10-0-0-27 nginx]$ sudo aws s3 cp s3://guhan3tierbucket/nginx.conf .
download: s3://guhan3tierbucket/nginx.conf to ./nginx.conf
[ec2-user@ip-10-0-0-27 nginx]$ sudo vi nginx.conf
[ec2-user@ip-10-0-0-27 nginx]$ sudo service nginx restart
Redirecting to /bin/systemctl restart nginx.service
[ec2-user@ip-10-0-0-27 nginx]$ sudo service nginx status
Redirecting to /bin/systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
    Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
    Active: active (running) since Tue 2023-09-12 21:14:13 UTC; 1min 32s ago
      Process: 26654 ExecStartPre=/usr/bin/m -f /run/nginx.pid (code=exited, status=0/SUCCESS)
      Process: 26661 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
      Process: 26669 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 26686 (nginx)
       Tasks: 2 (limit: 1114)
      Memory: 2.2M
        CPU: 59ms
       CGroup: /system.slice/nginx.service
           └─26686 "nginx: master process /usr/sbin/nginx"
              ├─26687 "nginx: worker process"

Sep 12 21:14:13 ip-10-0-0-27.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal nginx[26661]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal nginx[26661]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-0-27 nginx]$
```

- Check whether the Nginx got access by: `chmod -R 755 /home/ec2-user`

- Make to run when ever get boot run this command: `sudo chkconfig nginx on`

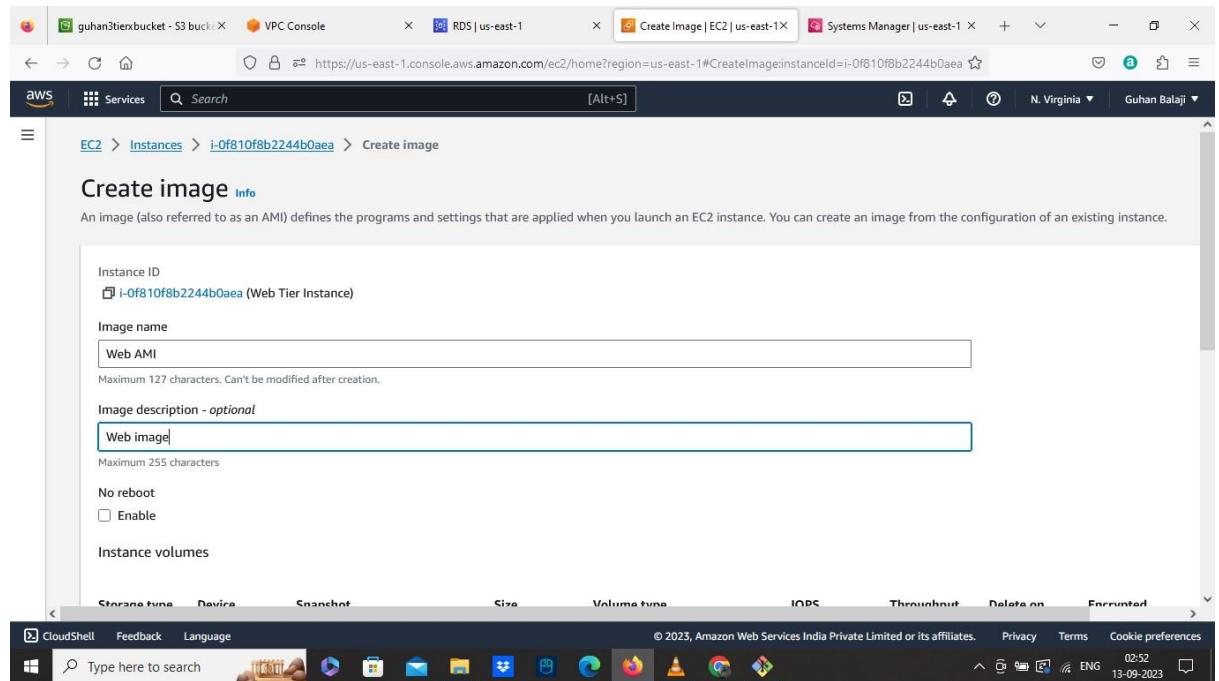


```
Session ID: root-02186c133ca55897c
Instance ID: i-0f810f8b2244b0aea
Terminate

Redirecting to /bin/systemctl restart nginx.service
[ec2-user@ip-10-0-0-27 nginx]$ sudo service nginx status
Redirecting to /bin/systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
    Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
    Active: active (running) since Tue 2023-09-12 21:14:13 UTC; 1min 32s ago
      Process: 26654 ExecStartPre=/usr/bin/m -f /run/nginx.pid (code=exited, status=0/SUCCESS)
      Process: 26661 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
      Process: 26669 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 26686 (nginx)
       Tasks: 2 (limit: 1114)
      Memory: 2.2M
        CPU: 59ms
       CGroup: /system.slice/nginx.service
           └─26686 "nginx: master process /usr/sbin/nginx"
              ├─26687 "nginx: worker process"

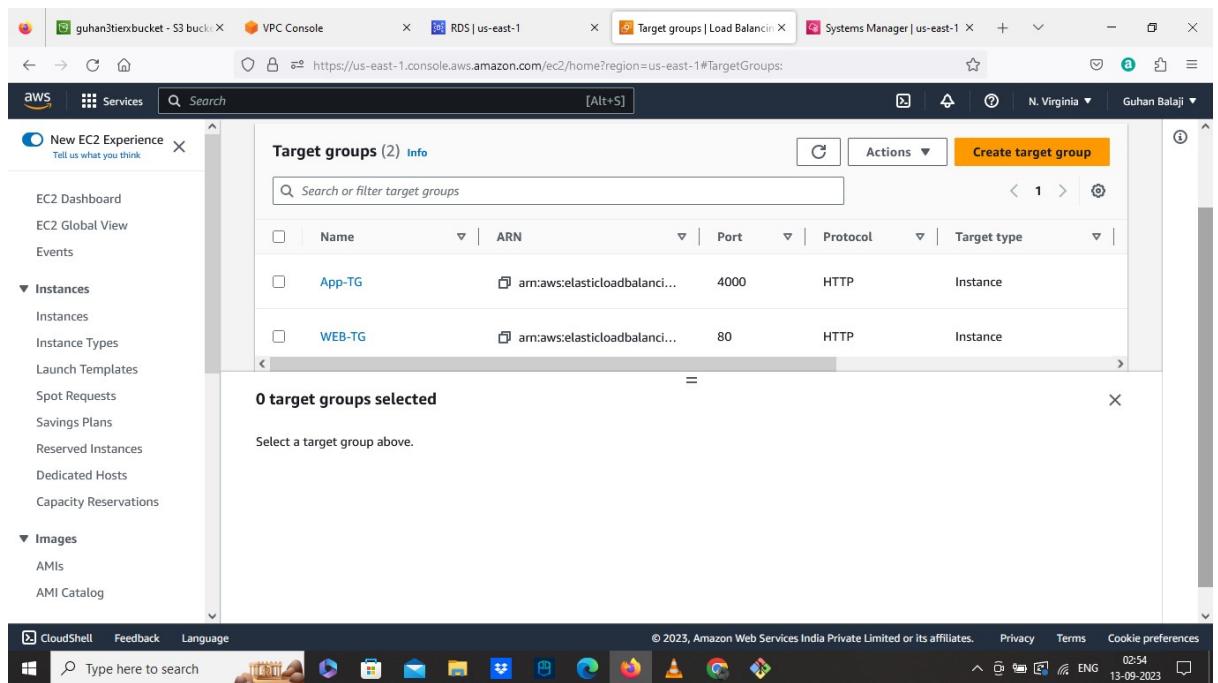
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal nginx[26661]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal nginx[26661]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 12 21:14:13 ip-10-0-0-27.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-0-27 nginx]$ sudo chkconfig nginx on
Note: Forwarding request to 'systemctl enable nginx.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-0-27 nginx]$
```

- Create an Image AMI from our WEB instance



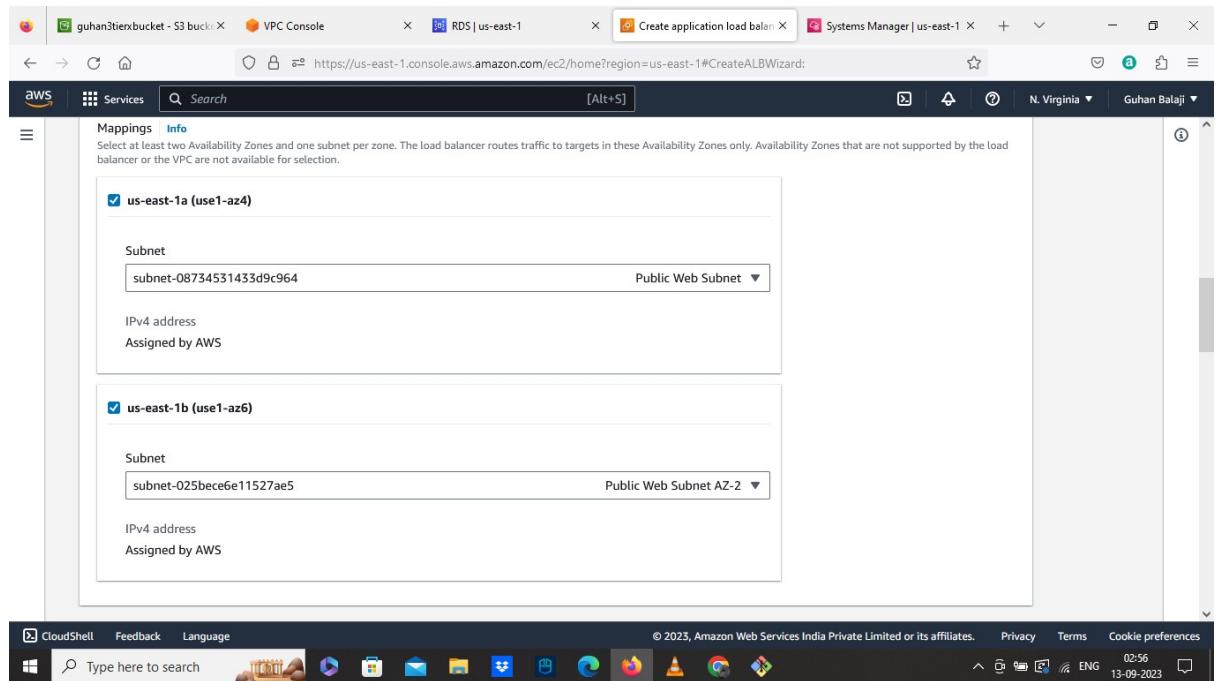
• Create Target Groups

- ➔ Select The type as Instances
- ➔ Give the name to our Target Group
- ➔ Select The port to 80
- ➔ Select our custom VPC



- Create Load balancer

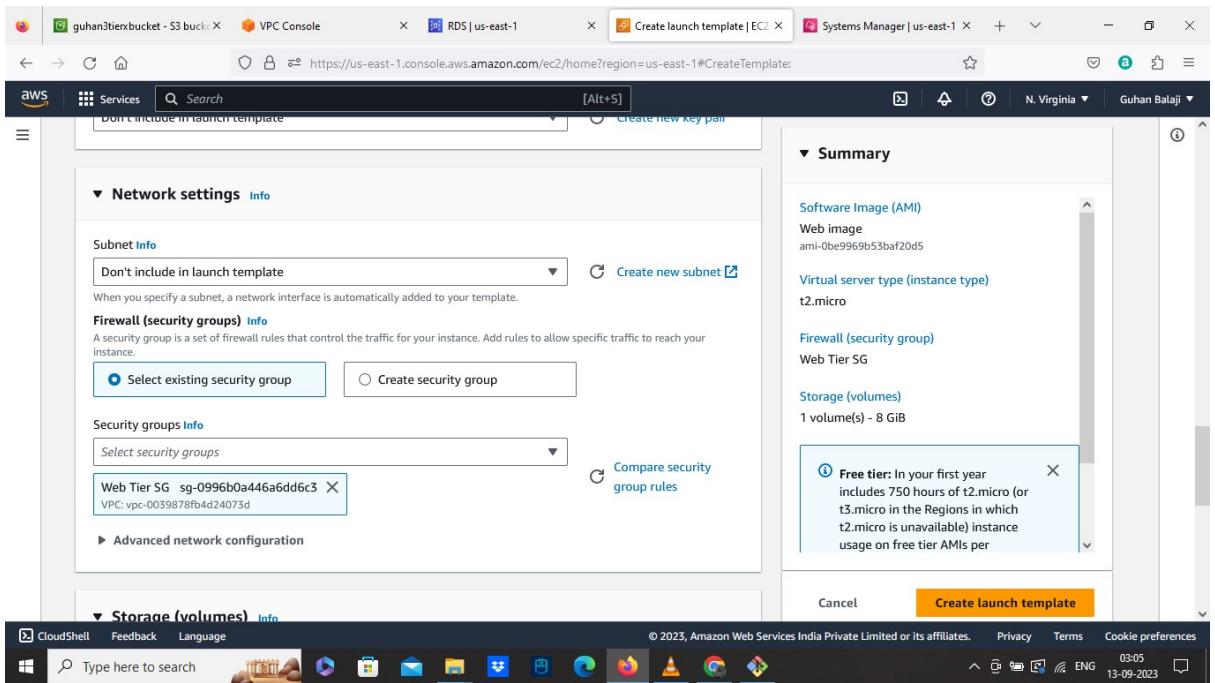
- ➔ Choose the scheme to Internet Facing
- ➔ Select our custom VPC
- ➔ In Mappings Select two public web subnets az1, az2
- ➔ Select the existing security group which is Internet Facing External Load balancer
- ➔ In Listeners and routing select port as 80 and forward to Web Target Group we created.



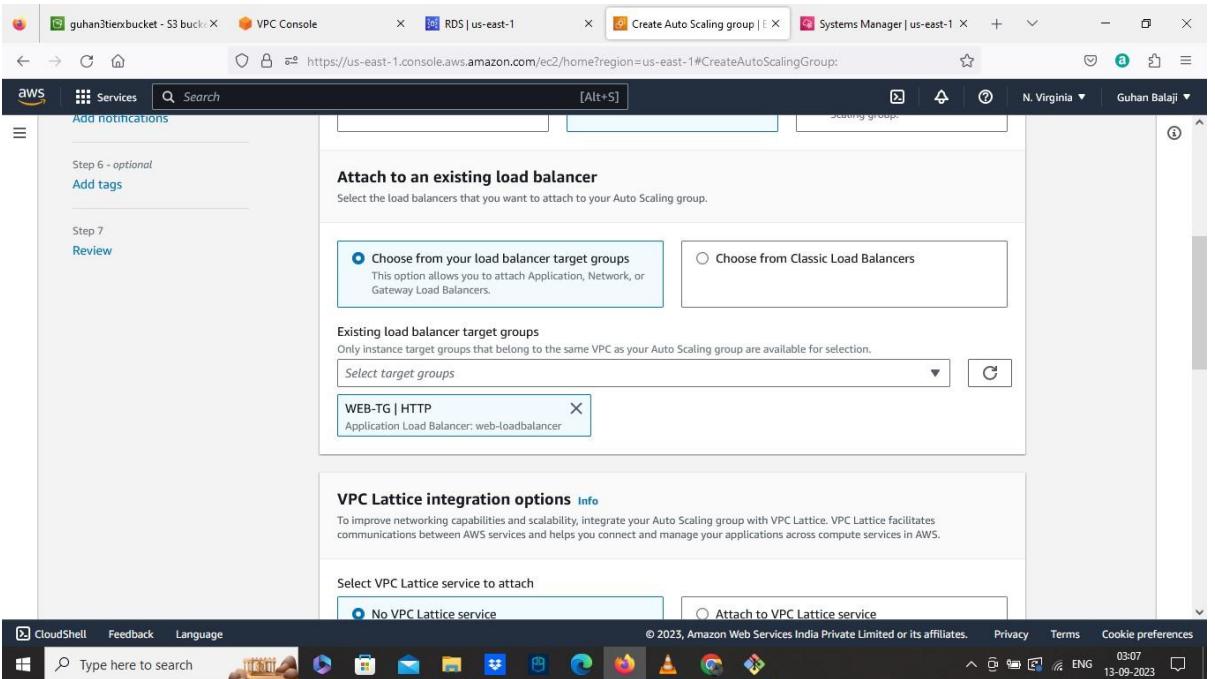
The screenshot shows the AWS CloudFront console with the 'Create new distribution' wizard. The 'Distribution Settings' tab is active. Under 'Origins', there is one origin pointing to an S3 bucket named 'guhan3tierbucket'. The 'Behaviors' section contains two entries: one for 'HTTP' and one for 'HTTPS'. The 'HTTPS' behavior is set to use a pre-signed certificate.

- Create the Launch Template
- Select our Web instance AMI from MY AMI
- Choose Instance type as free tier
- Select the existing security group which is Web tier SG
- Select our IAM role

The screenshot shows the AWS Lambda console with the 'Create new function' wizard. The 'Basic' tab is selected. In the 'Function code' section, the code for the Lambda function is displayed in a code editor. The code is a simple Hello World function written in Python.



- Create Autoscaling group
 - ➔ Give a name to the ASG
 - ➔ Select our Web Launch Template
 - ➔ Select our Custom VPC
 - ➔ Select the two public web subnets az1,az2
 - ➔ Then attach to existing load balancer we created for WEB
 - ➔ Select the WEB Target group
 - ➔ Change the value to 2 for group size.



The screenshot shows the AWS Load Balancers console. The left sidebar includes sections for AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), and Auto Scaling (Auto Scaling Groups). The main content area displays a table titled "Load balancers (2)" with columns for Name, DNS name, State, VPC ID, and Availability Zones. Two entries are listed:

Name	DNS name	State	VPC ID	Availability Zones
App-Loadbalancer	internal-App-Loadbalancer...	Active	vpc-0039878fb4d24073d	2 Availability Zones
web-loadbalancer	web-loadbalancer-149538...	Active	vpc-0039878fb4d24073d	2 Availability Zones

Below the table, a message says "0 load balancers selected" and "Select a load balancer above."

The screenshot shows the AWS Instances console. The left sidebar includes sections for New EC2 Experience (Tell us what you think), EC2 Dashboard, EC2 Global View, Events, Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area displays a table titled "Instances (6) Info" with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DN. Six instances are listed, all in a "Running" state:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
Web Tier Insta...	i-0fb10f8b2244b0aea	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
-	i-0884cc8b3af2b66e	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
-	i-0b0b1e7218be5aa5b	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
-	i-0e737a0525db1ba1	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	-
-	i-0e68eb86ffaf3a6b	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	-
App Tier Insta...	i-00c13fac0f501f112	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-

Below the table, a message says "Select an instance".

- Now Launch the website by copying the DNS name of WEB load balancer and paste into the browser

Screenshot of the AWS Cloud Console showing the Load Balancers page. The sidebar navigation includes:

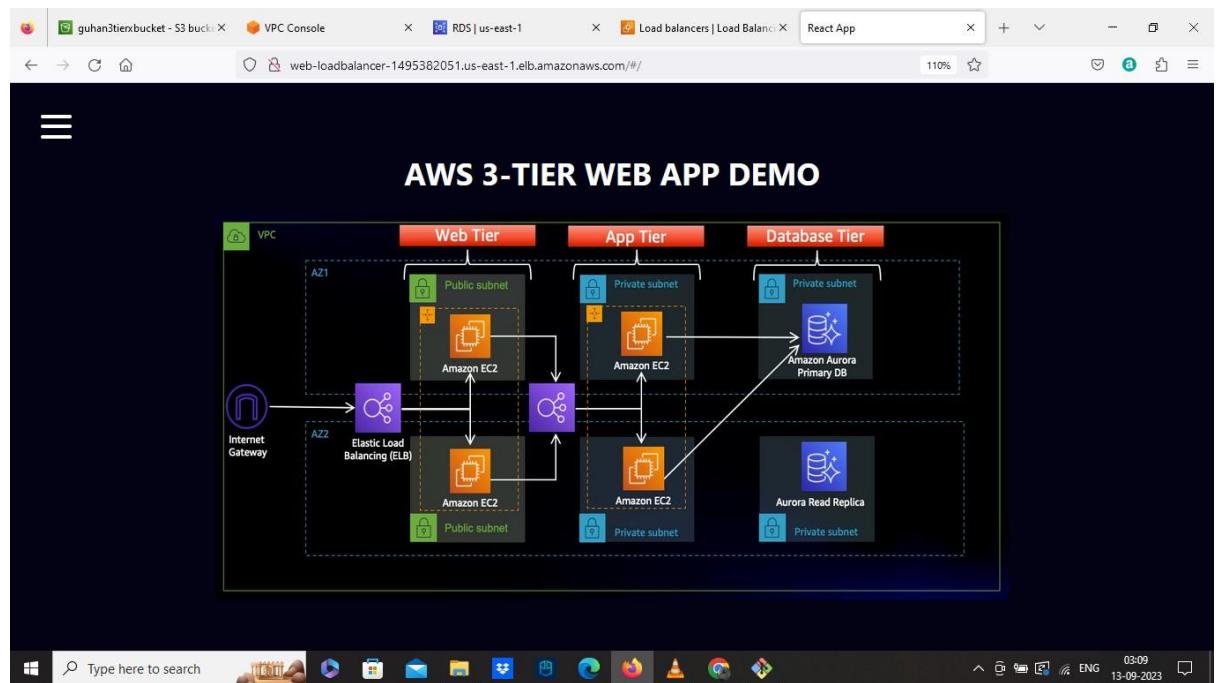
- AMI Catalog
- Elastic Block Store (Volumes, Snapshots, Lifecycle Manager)
- Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces)
- Load Balancing (selected), with sub-options: Load Balancers (2) and Target Groups.
- Auto Scaling (Auto Scaling Groups)

The main content area displays two load balancers:

Name	DNS name	State	VPC ID	Availability Zones
App-Loadbalancer	app-loadbalancer-1495382051.us-east-1.elb.amazonaws.com	Active	vpc-0039878fb4d24073	2 Availability Zones
web-loadbalancer	web-loadbalancer-1495382051.us-east-1.elb.amazonaws.com	Active	vpc-0039878fb4d24073	2 Availability Zones

A tooltip "DNS name copied" is shown over the first load balancer's DNS name field. A modal window at the bottom says "0 load balancers selected" and "Select a load balancer above." The browser status bar shows "CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 03:08 13-09-2023".

• OUTPUT:



AURORA DATABASE DEMO PAGE

ID	AMOUNT	DESC
1	400	groceries
2	500	accessories

DEL

Type here to search

03:11 13-09-2023

X

AURORA DATABASE DEMO PAGE

DEL

ID	AMOUNT	DESC
1	400	groceries
2	500	accessories

HOME

DB DEMO

web-loadbalancer-1495382051.us-east-1.elb.amazonaws.com/#/db

Type here to search

03:11 13-09-2023

The screenshot shows a web browser window with multiple tabs open, including S3 bucket, VPC Console, RDS | us-east-1, Load balancers | Load Balancer, and React App. The main content area displays a dark-themed application titled "AURORA DATABASE DEMO PAGE".

The application interface includes:

- A sidebar with icons for "HOME" (house) and "DB DEMO" (document).
- A table with the following data:

ID	AMOUNT	DESC
7	400	accessories
8	0	
9	300	cosmetics
10	0	
11	600	medicines
12	100	Guhan Pass
13	300	Subscriptions

An "ADD" button is located above the table, and a "DEL" button is located to its right.

The browser's taskbar at the bottom shows the Windows Start button, a search bar, and various pinned icons for Microsoft Office applications (Word, Excel, etc.) and other software.

ARCHITECTURE OF THE PROJECT:

