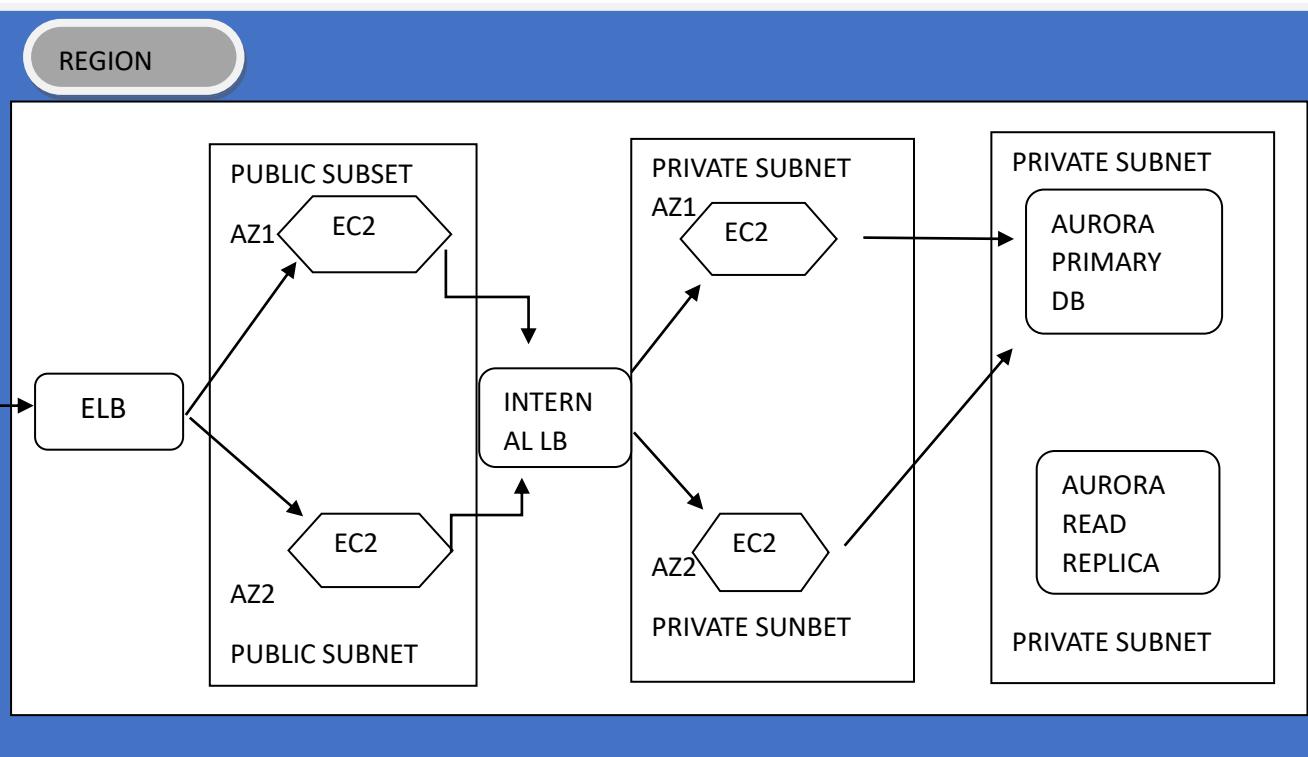


# Three Tier Architecture on AWS

A.Isac jabaraj

## ARCHITECTURE

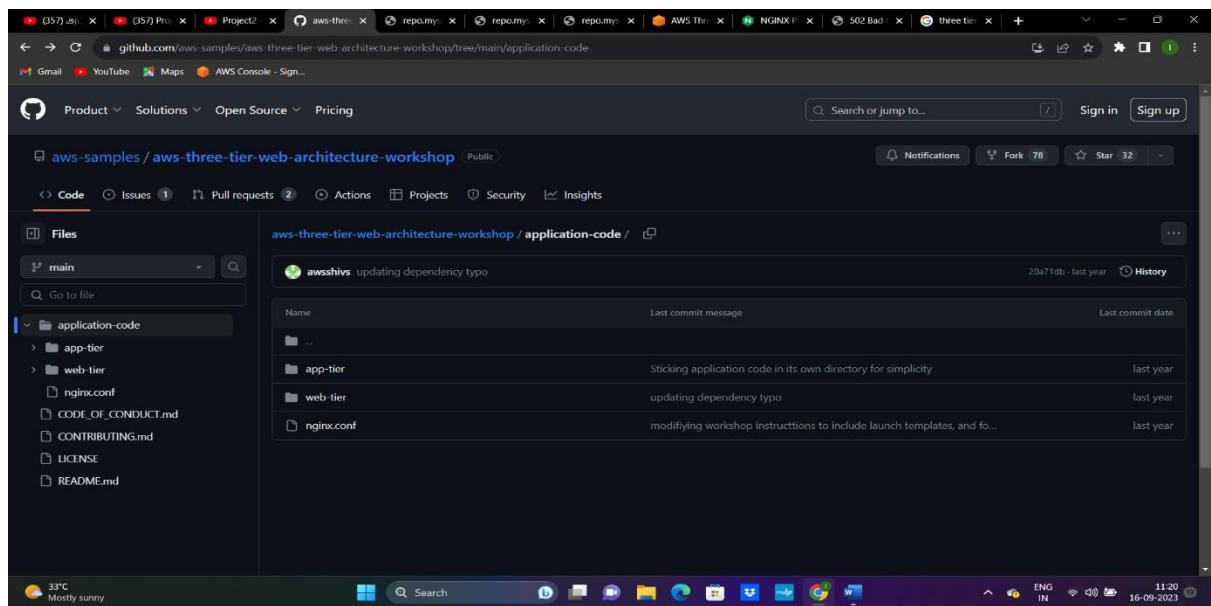


## Introduction

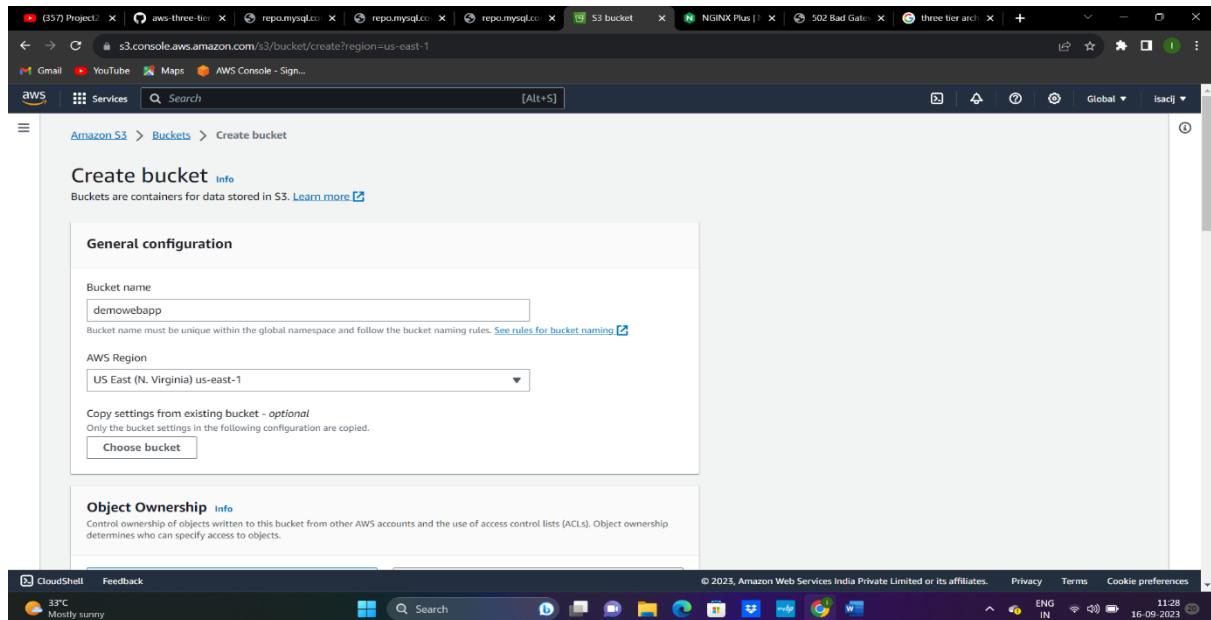
Three tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application.

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.

## STEP 0 :Download the App tier and web tier files from the Github.



## Step 1: Create S3 Bucket.



## Step 2 : Create IAM Role in EC2 and Add Permission

### AmazonSSMManagedInstanceCore

### AmazonS3ReadOnlyAccess

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and various management sections like 'User groups', 'Users', 'Roles', 'Policies', etc. The main area is titled 'Summary' and shows details such as 'Creation date: September 13, 2023, 07:47 (UTC+05:30)', 'ARN: arn:aws:iam::823119645222:role/demo-ec2', and 'Last activity: 15 minutes ago'. Below this, the 'Permissions' tab is active, showing a table of attached policies:

Policy name	Type	Attached entities
AmazonS3ReadOnlyAccess	AWS managed	1
AmazonSSMManagedInstanceCore	AWS managed	1

## Step 3: Create VPC

The screenshot shows the 'Create VPC' page in the AWS VPC Manager. The 'VPC settings' section is open, showing the following configuration:

- Resources to create:** VPC only (selected)
- Name tag - optional:** awsdemo
- IPv4 CIDR block:** 10.0.0.0/24
- IPv6 CIDR block:** No IPv6 CIDR block

At the top, there are navigation links: 'VPC' > 'Your VPCs' > 'Create VPC'. The bottom of the page includes standard AWS footer links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

**Step 4: Create subnets** 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.

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with options like '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, Peering connections) and 'Security'. The main area displays a table titled 'Subnets (12)' with columns: Name, Subnet ID, State, VPC, and IPv4 CIDR. The table lists 12 subnets, half of which are in 'public' VPCs and half in 'private' VPCs, distributed across two availability zones. Below the table is a section titled 'Select a subnet' with a search bar and a list of subnet names.

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

The screenshot shows the 'Create subnet' page. At the top, it says 'VPC > Subnets > Create subnet'. The page has two main sections: 'VPC' and 'Subnet settings'. In the 'VPC' section, 'VPC ID' is set to 'Create subnets in this VPC.' with 'vpc-0e3328fc51249ff3b (awsdemo)' selected. Under 'Associated VPC CIDRs', 'IPv4 CIDRs' is listed as '10.0.0.0/16'. In the 'Subnet settings' section, it says 'Specify the CIDR blocks and Availability Zone for the subnet.' Under 'Subnet 1 of 1', 'Subnet name' is set to 'private Az1'. A note below says 'The name can be up to 256 characters long.' The bottom of the screen shows a Windows taskbar with various icons and a weather widget.

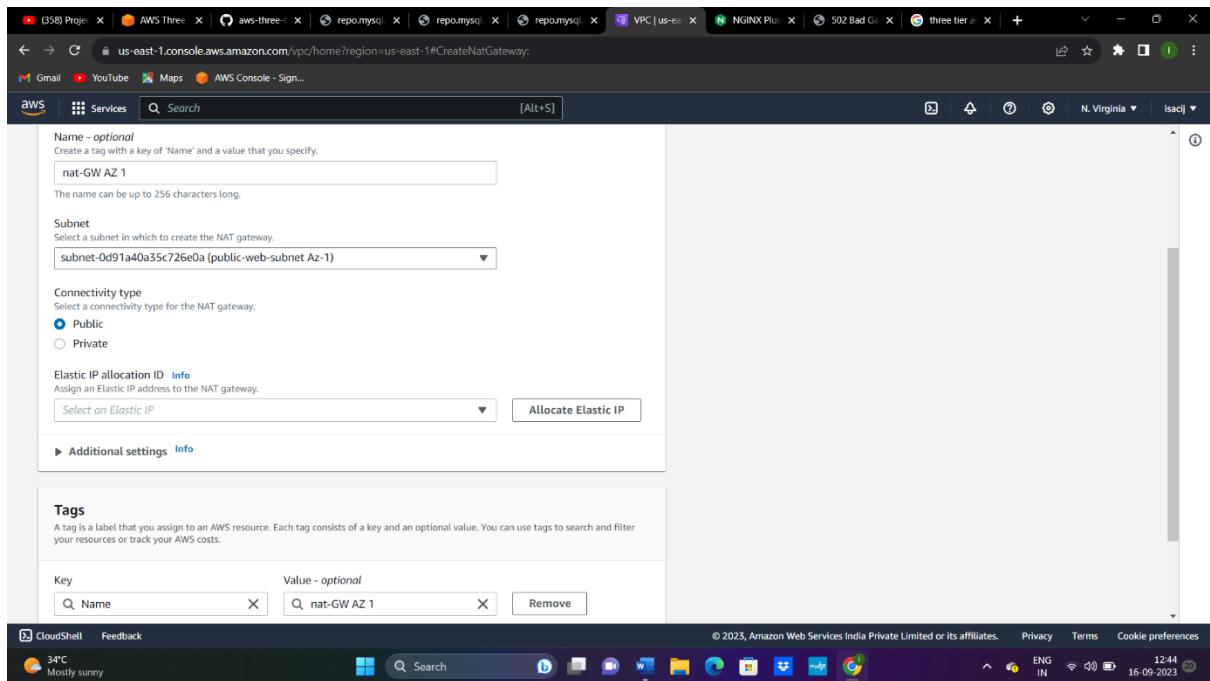
## Step 5: Create INTERNET GATEWAY

The screenshot shows the 'Create internet gateway' page in the AWS VPC console. The 'Internet gateway settings' section contains a 'Name tag' input field with the value 'three-tier'. Below it is a 'Tags - optional' section with a single tag 'Name: three-tier'. The page includes a status bar at the bottom showing weather information (34°C, Hot weather), system icons, and navigation links.

## Attach to the VPC

The screenshot shows the 'igw-064419ffcf7686f9d / three-tier-demo-igw' details page. The 'Actions' menu on the right includes options like 'Attach to VPC', 'Detach from VPC', 'Manage tags', and 'Delete'. The 'Details' tab shows the internet gateway ID, state (Attached), VPC ID, and owner. The 'Tags' section lists a single tag 'Name: three-tier-demo-igw'. The left sidebar shows the VPC dashboard and various network components like subnets and route tables. The page includes a status bar at the bottom showing weather information (34°C, Mostly sunny), system icons, and navigation links.

## Step 6: Create NAT GATEWAY for the Both zone



And do the same processes for the other Zone.

## Step 7: Create two Route table for Public,Private.

Edit the routes and add one more to the routes .

- In public we have to give target igw ,
- In private we have to give Nat .
- Subnets associations we sperate public and private according through the zone but in public we add two public zone in one subnet association ,In private according through that subnet association.

**Create route table** Info

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

**Route table settings**

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

VPC  
The VPC to use for this route table.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Q Name"/>	<input type="text" value="public routetable"/>
<input type="button" value="Remove"/>	
<input type="button" value="Add new tag"/>	
You can add 49 more tags.	

CloudShell Feedback 34°C Mostly sunny Search Privacy Terms Cookie preferences © 2023, Amazon Web Services India Private Limited or its affiliates. ENG IN 11:00 16-09-2023

**Route tables (1/4) Info**

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC
<input checked="" type="checkbox"/> publicRoutetable	rtb-05d7ab840f529a15a	2 subnets	-	No	vpc-0e3328fc51249ff3b   i
<input type="checkbox"/> -	rtb-00b0ffd7c2e8e2822	-	-	Yes	vpc-0e3328fc51249ff3b   i
<input type="checkbox"/> -	rtb-084c83445d8aeb885	-	-	Yes	vpc-090d9243360e94654
<input type="checkbox"/> PrivateRoute Table -AZ1	rtb-03bf7b1ce485a7c54	subnet-0153799d2d0dfc...	-	No	vpc-0e3328fc51249ff3b   i

**Routes (2)**

Destination	Target	Status	Propagated
0.0.0.0/0	igw-064419ffc7f766f9d	Active	No
10.0.0.16	local	Active	No

CloudShell Feedback 34°C Partly sunny Search Privacy Terms Cookie preferences © 2023, Amazon Web Services India Private Limited or its affiliates. ENG IN 11:06 16-09-2023

**Route tables (1/4) Info**

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC
<input checked="" type="checkbox"/> publicRoutetable	rtb-05d7ab840f529a15a	2 subnets	-	No	vpc-0e3328fc51249ff3b   i
<input type="checkbox"/> -	rtb-00b0ffd7c2e8e2822	-	-	Yes	vpc-0e3328fc51249ff3b   i
<input type="checkbox"/> -	rtb-084c83445d8aeb885	-	-	Yes	vpc-090d9243360e94654
<input type="checkbox"/> PrivateRoute Table -AZ1	rtb-03bf7b1ce485a7c54	subnet-0153799d2d0dfc...	-	No	vpc-0e3328fc51249ff3b   i

**Subnet associations**

Details	Routes	Subnet associations	Edge associations	Route propagation	Tags
<b>Explicit subnet associations (2)</b>					
<input type="text" value="Find subnet association"/>					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR		
public-web-subnet Az-1	subnet-0d91a40a35c726e0a	10.0.0.0/24	-		
public-web-subnet Az-2	subnet-05e9d6ea47e46b47d	10.0.1.0/24	-		

CloudShell Feedback 34°C Partly sunny Search Privacy Terms Cookie preferences © 2023, Amazon Web Services India Private Limited or its affiliates. ENG IN 11:11 16-09-2023

Screenshot of the AWS VPC Route Tables page (Route tables (1/4)).

The table shows the following route tables:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
publicRoutetable	rtb-05d7ab840f329a15a	2 subnets	-	No	vpc-0e3328fc51249ff3b [1]
-	rtb-00b0ffd7c2e8e2822	-	-	Yes	vpc-0e3328fc51249ff3b [1]
-	rtb-084c83445d8aeb885	-	-	Yes	vpc-090d9243360e94654
<b>PrivateRoute Table -AZ1</b>	<b>rtb-03bf7b1ce485a7c54</b>	<b>subnet-0153799d2d0dfcfba</b>	-	No	vpc-0e3328fc51249ff3b [1]

The "Routes" section shows two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	nat-06a1bc4b7215f9438	Active	No
10.0.0.0/16	local	Active	No

Screenshot of the AWS VPC Route Tables page (Route tables (1/4)).

The table shows the same route tables as the previous screenshot.

The "rtb-03bf7b1ce485a7c54 / PrivateRoute Table -AZ1" details page is displayed. The "Subnet associations" tab is selected.

The "Explicit subnet associations (1)" table shows:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
private subnet Az-1	subnet-0153799d2d0dfcfba	10.0.2.0/24	-

## Step 8: Creating 5 security groups

The screenshot shows the AWS VPC Management Console with the URL <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#SecurityGroups>. The page displays a table titled "Security Groups (14) Info" with columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. The table lists 14 security groups, including "Internetfacing-lb-sg", "default", "GlobalAccelerator", "privateinstance-sg", "internal -lb-sg", and "DB-sg". The "Owner" column shows the same ID (823119645222) for all groups. The left sidebar shows the "Virtual private cloud" and "Security" sections.

First for public internet facing load balancer . after typing a name and description,Add an inbound to allow the http type traffic for Ip , source type IPV

Add one more for inbound type custom tcp and source type IPV6

The screenshot shows the AWS VPC Management Console with the URL <https://us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#SecurityGroups>. The page displays a table titled "Security Groups (1/14) Info" with columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. The table lists 14 security groups, including "Internetfacing-lb-sg", "default", "GlobalAccelerator", "privateinstance-sg", "internal -lb-sg", and "DB-sg". The "Owner" column shows the same ID (823119645222) for all groups. Below the table, there is a section titled "Inbound rules (2)" with a table showing two rules: one for IPv6 (Custom TCP, Port 0) and one for IPv4 (HTTP, Port 80). The "Owner" column shows the same ID (823119645222) for both rules. The left sidebar shows the "Virtual private cloud" and "Security" sections.

Second for the web tier sg , add an inbound to allow the http type traffic , source type give custom and (select sg webtier in the type ) , add one more in an inbound to allow the http and source type my IP

The screenshot shows the AWS VPC Security Groups page. In the main table, there is a row for 'sg-029c5b1bea716f259' named 'webtier-sg'. In the 'Inbound rules' section, two rules are listed:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0c7c7ac97047167ae	IPv4	HTTP	TCP	80	110.224.84.188/32	
sgr-045e4fc5b8c6ba7f1	-	HTTP	TCP	80	sg-016dd20a2d7589	

Third security groups for internal load balancer , add an inbound to allow http type traffic , source type be custom and target sg webtier.

The screenshot shows the AWS VPC Security Groups page. In the main table, there is a row for 'sg-0efbace1b0766a3c07' named 'internal-ib-sg'. In the 'Inbound rules' section, one rule is listed:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0a8f1de6ef7c6e42	-	HTTP	TCP	80	sg-029c5b1bea716f2	

Fourth security groups for Privateinstance-sg , Add an inbound type custom TCP and Port range 4000, source .Type custom and target sg internal lb. Add other another inbound type custom TCP and port range 4000, source type give myIP

The screenshot shows the AWS VPC Security Groups console. The main table lists 14 security groups, including 'privateinstance-sg' which is selected. Below the table, the 'Inbound rules (1/1)' section shows one rule:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
sgr-0551c882c007aaea4	IPv4	Custom TCP	TCP	4000		110.224.84.188/32

Fifth security group for DB-sg , add an inbound type MYSQL/Aurora. Source type custom and target sg privateinstance.

The screenshot shows the AWS VPC Security Groups console. The main table lists 14 security groups, including 'DB-sg' which is selected. Below the table, the 'Inbound rules (1/1)' section shows one rule:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
sgr-02f4908cb902e513c	-	MySQL/Aurora	TCP	3306		sg-06245c924e2d3b5

## Step 9: Create Database in Amazon RDS

Go to subnet group and create DB subnet group.

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

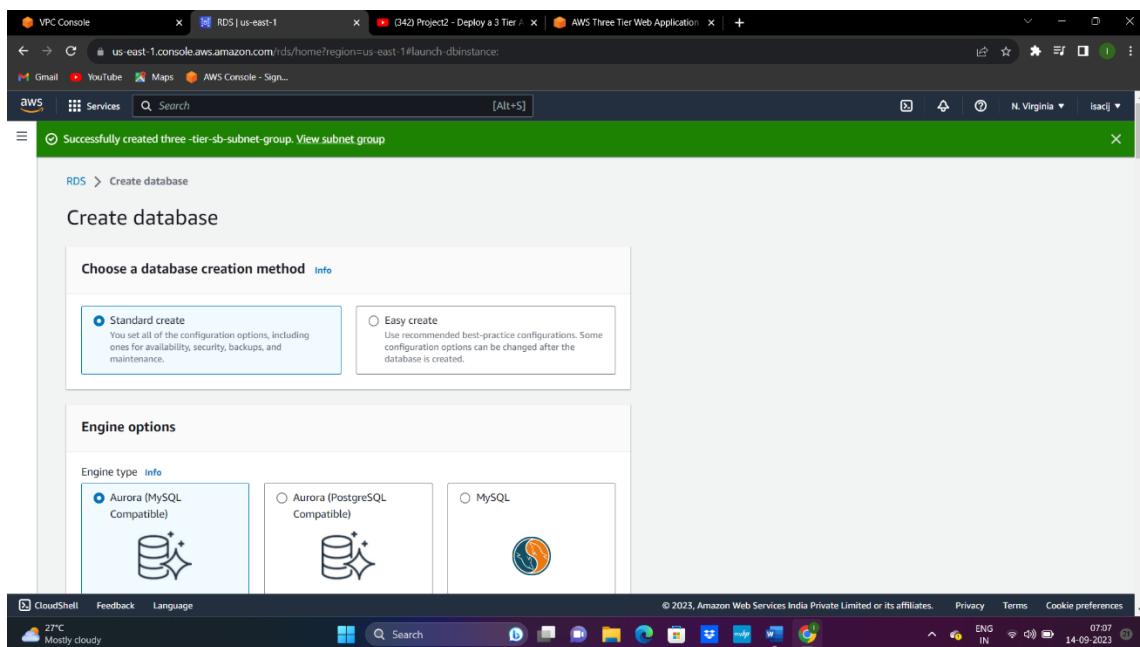
The screenshot shows the 'Create DB subnet group' page in the AWS RDS console. In the 'Subnet group details' section, the 'Name' field contains 'three-tier-subnet-group'. The 'Description' field also contains 'three-tier-subnet-group'. The 'VPC' dropdown is set to 'awsdemo (vpc-0e3328fc51249ff3b)'. Below this, the 'Add subnets' section shows two selected Availability Zones: 'us-east-1a' and 'us-east-1b'. A note at the bottom of this section states: 'For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.'

The screenshot shows the 'Add subnets' page in the AWS RDS console. Under 'Availability Zones', 'us-east-1a' and 'us-east-1b' are selected. Under 'Subnets', two subnets are selected: 'subnet-0d91a40a35c726e0a (10.0.0.0/24)' and 'subnet-05e9d6ea47e46b47d (10.0.1.0/24)'. At the bottom of the 'Subnets selected' section, there is a table:

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-0d91a40a35c726e0a	10.0.0.0/24
us-east-1b	subnet-05e9d6ea47e46b47d	10.0.1.0/24

And next do to Database create it by following this Points.

- 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
- Turn OFF the performance insights and create database



VPC Console RDS | us-east-1 (342) Project2 - Deploy a 3 Tier AWS Three Tier Web Application

Gmail YouTube Maps AWS Console - Sign...

aws Services Search [Alt+S]

Memory optimized classes (includes r classes)  
Burstable classes (includes t classes)

db.r6g.2xlarge  
8 vCPUs 64 GB RAM Network: 4.750 Mbps

Include previous generation classes

**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 fast failover and high availability.  
 Don't create an Aurora Replica

**Connectivity** [Info](#)

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

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

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

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 27°C Mostly cloudy 07:10 14-09-2023

VPC Console RDS | us-east-1 (342) Project2 - Deploy a 3 Tier AWS Three Tier Web Application

Gmail YouTube Maps AWS Console - Sign...

aws Services Search [Alt+S]

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.

**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.

**Virtual private cloud (VPC)** [Info](#)  
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

awsdemo (vpc-0e3528fc51249ff3b)  
6 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

**DB subnet group** [Info](#)  
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.

three-tier-sb-subnet-group  
2 Subnets, 2 Availability Zones

**Public access** [Info](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 27°C Mostly cloudy 07:11 14-09-2023

VPC Console RDS | us-east-1 (342) Project2 - Deploy a 3 Tier AWS Three Tier Web Application

Gmail YouTube Maps AWS Console - Sign...

aws Services Search [Alt+S]

Show versions that support Serverless v2  
Offers instance scaling for even the most demanding workloads.

Available versions (16/17) [Info](#)  
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.  
database-1

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 27°C Mostly cloudy 07:07 14-09-2023

The screenshot shows the AWS RDS console interface for creating a new database instance. The current step is 'Database authentication'. The 'Password authentication' option is selected, which authenticates using database passwords. Other options like 'Password and IAM database authentication' and 'Password and Kerberos authentication' are also listed. Below the authentication section, there's a 'Monitoring' section with a checkbox for 'Turn on Performance Insights' and a link to 'Enhanced Monitoring'. At the bottom, there's a 'Additional configuration' section with a 'Next Step' button. The browser status bar at the bottom indicates it's 07:13 on 14-09-2023.

RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

No  
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.

**VPC security group (firewall) Info**  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

**Existing VPC security groups**  
Choose one or more options ▾  
DB-sg X

**RDS Proxy**  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.  
 Create an RDS Proxy Info  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS RDS pricing](#).

▶ Additional configuration

**Database authentication**

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

CloudShell Feedback Language 27°C Mostly cloudy Search ENG IN 07:13 14-09-2023

Databases (3)

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

Endpoints (2)

Endpoint name	Status	Type	Port
database-1.cluster-cusr2sf0wj4g.us-east-1.rds.amazonaws.com	Available	Writer	3306
database-1.cluster-ro-cusr2sf0wj4g.us-east-1.rds.amazonaws.com	Available	Reader	3306

## Step 10: 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
  - And Launch it.
- Connect to terminal
- Use Command sudo -su ec2-user
  - And Ping 8.8.8.8 to check the outside internet connectivity
- 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>
  - 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>.
  - Sudo yum install mysql.

After that use this commands for DB

- Use Command mysql
- Then Initiate your DB connection with your Aurora RDS writer endpoint. By the command: mysql -h database-1-instance-1.cusr2sf0wj4g.us-east-1.rds.amazonaws.com -u admin -p.
- And Enter the Password You Already created for the database.
- Create a database by using this command and name it :  
CREATE DATABASE webappdb;
- Use this command to check whether the database we created is available type it: SHOW DATABASES;
- Change to your database by using this webappdb;
- 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');
  - The data added by: SELECT \* FROM transactions;
  - Type exit to exit from the SQL.

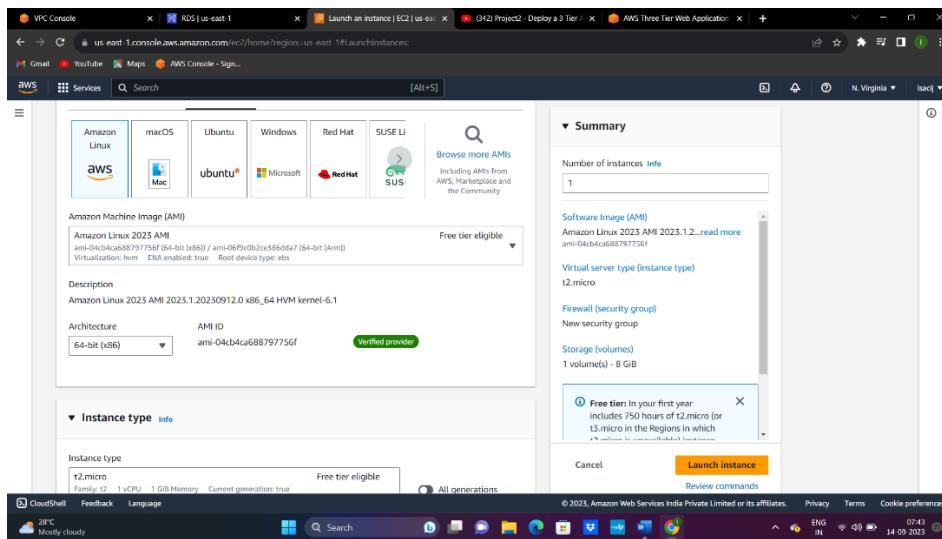
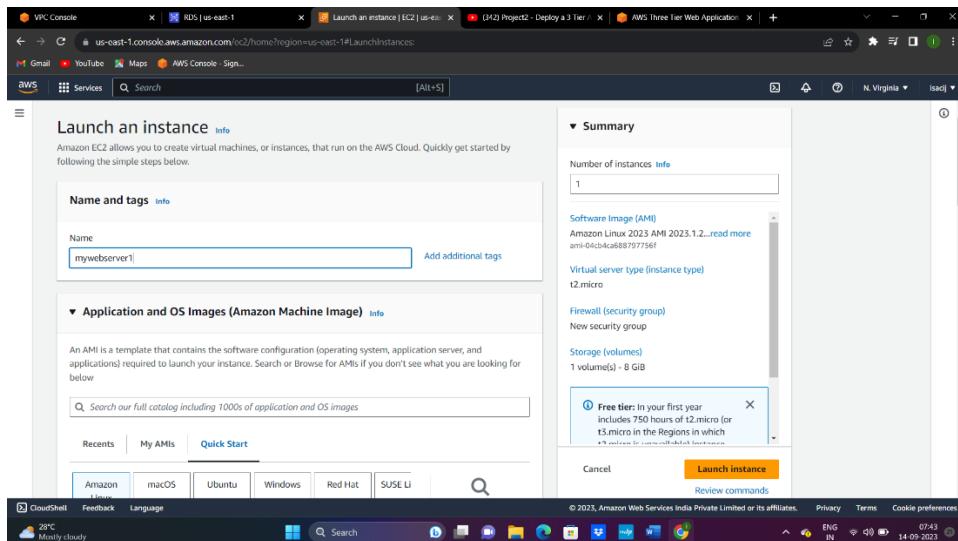
## Step 11: Go to S3 bucket we created

- 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.
- 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.
- And upload in the app tier folder and also for S3 bucket.

## Step 12: After this return to the connect terminal and follow this points

- Now we need to install components to run
- Install by running this command: curl -  
<https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh> |  
 bash source ~/.bashrc
- Running this command: nvm install 16 ,nvm use 16
- Install pm2,
- Use this command to install: npm install -g pm2
- Change the directory cd
- Now we need to download the code from the S3 bucket, so we need to use this command: aws s3 cp s3://demowebapp-111/app-tier/ app-tier – recursive.
- Add our bucket name in the command and run.
- Now download the entire folder ls -lrtr.
- Switch to app tier by using cd app-tier/
- Activate the pm2 we installed by using this command: pm2 start index.js.

- 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.
- Save : pm2 save
- Test the app health by :curl http://localhost:4000/health
- Test the database connection by using: curl  
http://localhost:4000/transcation
- After that app layer will be created.



VPC - required info

vpc-0e3328fc51240ff5b (awndemo)  
10.0.0.0/16

Subnet info

subnet-0154790d040dfcfa private subnet Az-1  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1a  
IP addresses available: 251 CIDR: 10.0.2.0/24

Q

subnet-0691a40a35c726dc public-web-subnet Az-1  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1a  
IP addresses available: 250 CIDR: 10.0.0.0/24

subnet-03760913b3956c5 private subnet Az-2  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1b  
IP addresses available: 250 CIDR: 10.0.3.0/24

subnet-05e956e04746d47d public-web-subnet Az-2  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1b  
IP addresses available: 250 CIDR: 10.0.1.0/24

subnet-091b116162bd0833 private DB-subnet Az-2  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1b  
IP addresses available: 250 CIDR: 10.0.5.0/24

subnet-0e7a02fd3d16c91 private DB-subnet Az-1  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1a  
IP addresses available: 250 CIDR: 10.0.4.0/24

subnet-015379930200dfcfa private subnet Az-1  
VPC vpc-0e3328fc51240ff5b Owner: 823119645222 Availability Zone: us-east-1b  
IP addresses available: 251 CIDR: 10.0.0.0/24

Security group rule 1 (TCP 22, 0.0.0.0/0)

Create new subnet

Remove

Summary

Number of instances: 1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2... read more  
ami-04cb4c688797756f

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

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 you've launched an instance.

Cancel Launch instance Review commands

CloudShell Feedback Language

ZFC Mostly cloudy

CloudShell Feedback Language

CloudShell Feedback Language

CloudShell Feedback Language

Auto-assign public IP: Info

Disable

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.

Create security group  Select existing security group

Common security groups info

Select security groups

privateinstance-sg sg-06245c924e2d3b558 X

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Advanced network configuration

Configure storage Info

Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

CloudShell Feedback Language

CloudShell Feedback Language

CloudShell Feedback Language

CloudShell Feedback Language

Advanced details Info

Purchasing option Info

Request Spot Instances

Domain join directory Info

Select Create new directory

IAM instance profile Info

Select Create new IAM profile

demo-ec2 amavasiam:823119645222:instance-profile/demo-ec2

isac amavasiam:823119645222:instance-profile/isac

test amavasiam:823119645222:instance-profile/test

you amavasiam:823119645222:instance-profile/you

Select

Summary

Number of instances: 1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2... read more  
ami-04cb4c688797756f

Virtual server type (instance type)

t2.micro

Firewall (security group)

privateinstance-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 you've launched an instance.

Cancel Launch instance Review commands

CloudShell Feedback Language

CloudShell Feedback Language

CloudShell Feedback Language

CloudShell Feedback Language

```

sh: 5:25 sudo -su ec2-user
[ec2-user@ip-10-0-2-100 bin]$ ping 8.8.8.8
Session ID: root-0f6b17c4bac789eb
Instance ID: i-08da4677cddeaba389
Terminate

```

sh: 5:25 sudo -su ec2-user
[ec2-user@ip-10-0-2-100 bin]\$ ping 8.8.8.8

```

sh: 5:25 sudo -su ec2-user
[ec2-user@ip-10-0-2-100 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=110 time=2.10 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=110 time=1.33 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=110 time=1.33 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=110 time=1.32 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=110 time=1.35 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=110 time=1.29 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=110 time=1.82 ms
...
-- 8.8.8.8 ping statistics --
7 packets transmitted, 0 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.392/1.506/2.097/0.295 ms
[ec2-user@ip-10-0-2-100 bin]$ 

```

```

sh: 5:25 sudo -su ec2-user
[ec2-user@ip-10-0-2-100 bin]$ ping 8.8.8.8
Session ID: root-0f6b17c4bac789eb
Instance ID: i-08da4677cddeaba389
Terminate

```

sh: 5:25 sudo -su ec2-user
[ec2-user@ip-10-0-2-100 bin]\$ ping 8.8.8.8

```

mysql57-community-release-el7-11.noarch.rpm:4: Permission denied
Cannot write to 'mysql57-community-release-el7-11.noarch.rpm' (Permission denied).
[ec2-user@ip-10-0-2-100 bin]$ sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
--2023-09-14 05:10:43-- https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 96.7.17.219, 2600:1408:c400:188c:2e31, 2600:1408:c400:188c:2e31
Connecting to dev.mysql.com (dev.mysql.com)|96.7.17.219|: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' [4]

mysql57-community-release-el7-11.noarch.rpm 100%[=====] 25.08K --=KB/s in 0.006s

2023-09-14 05:10:44 (4.19 MB/s) - 'mysql57-community-release-el7-11.noarch.rpm' saved [25680/25680]

[ec2-user@ip-10-0-2-100 bin]$ sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
curl: (22) The requested URL returned error: 404
error: http://repo.mysql.com/RPM-GPG-KEY-mysql-2022: import read failed(2).
[ec2-user@ip-10-0-2-100 bin]$ sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
[ec2-user@ip-10-0-2-100 bin]$ sudo yum install https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
MySQL Connectors Community
MySQL Tools Community
MySQL 5.7 Community Server
mysql57-community-release-el7-11.noarch.rpm
Package mysql57-community-release-el7-11.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-2-100 bin]$ sudo yum install mysql

```

VPC Console | RDS us-east-1 | Connect to instance | Systems Manager | (342) Project | repo.mysql.com | repo.mysql.com | AWS Three Tier | Downloads | +

Last metadata expiration check: 0:00:50 ago on Thu Sep 14 05:13:00 2023.  
Dependencies resolved.

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

Transaction Summary  
Install 4 Packages

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 3.2 MB/s | 313 KB 00:00  
(2/4): mysql-community-libs-5.7.43-1.el7.x86\_64.rpm 2.9 MB/s | 322 KB 00:00  
(3/4): ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86\_64.rpm 1.7 MB/s | 322 KB 00:00  
(4/4): mysql-community-client-5.7.43-1.el7.x86\_64.rpm 60 MB/s | 31 MB 00:00

Total  
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  
32°C Mostly cloudy 10:44 14-09-2023

VPC Console | RDS us-east-1 | Connect to instance | Systems Manager | (342) Project | repo.mysql.com | repo.mysql.com | AWS Three Tier | Downloads | +

Last metadata expiration check: 0:00:50 ago on Thu Sep 14 05:13:00 2023.  
Dependencies resolved.

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

Transaction Summary  
Install 4 Packages

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 3.2 MB/s | 313 KB 00:00  
(2/4): mysql-community-libs-5.7.43-1.el7.x86\_64.rpm 2.9 MB/s | 322 KB 00:00  
(3/4): ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86\_64.rpm 1.7 MB/s | 322 KB 00:00  
(4/4): mysql-community-client-5.7.43-1.el7.x86\_64.rpm 60 MB/s | 31 MB 00:00

Total  
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  
32°C Mostly cloudy 10:44 14-09-2023

VPC Console | RDS us-east-1 | Connect to instance | Systems Manager | (342) Project | repo.mysql.com | repo.mysql.com | AWS Three Tier | Downloads | +

Last metadata expiration check: 0:00:50 ago on Thu Sep 14 05:13:00 2023.  
Dependencies resolved.

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

Transaction Summary  
Install 4 Packages

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 3.2 MB/s | 313 KB 00:00  
(2/4): mysql-community-libs-5.7.43-1.el7.x86\_64.rpm 2.9 MB/s | 322 KB 00:00  
(3/4): ncurses-compat-libs-6.2-4.20200222.amzn2023.0.4.x86\_64.rpm 1.7 MB/s | 322 KB 00:00  
(4/4): mysql-community-client-5.7.43-1.el7.x86\_64.rpm 60 MB/s | 31 MB 00:00

Total  
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  
32°C Mostly cloudy 11:02 14-09-2023

```

Session ID: root-049b02b12f0fafe1 Instance ID: i-08da4677cdeaba389
Terminate

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 'database' at line 1
1 mysql>
2 mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| 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 (0.01 sec)

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

mysql>

```

22°C Mostly cloudy 14-09-2023

```

Session ID: root-049b02b12f0fafe1 Instance ID: i-08da4677cdeaba389
Terminate

+-----+
| Database |
+-----+
| information_schema |
| mysql |
| 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 (0.01 sec)

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

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

mysql>

```

32°C Mostly cloudy 14-09-2023

```

Session ID: root-03adfc1e4e6cd643 Instance ID: i-08da4677cdeaba389
Terminate

[Install]
WantedBy=multi-user.target

[Service]
Type=forking
User=ec2-user
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment="PATH=/home/ec2-user/.nvm/versions/node/v16.20.2/bin:/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/bin:/usr/sbin:/bin:/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:/sbin:/lib/libc:/lib64/libc:/sbin:/bin:/home/ec2-user/.nvm/versions/node/v16.20.2/bin:/bin:/usr/local/bin:/usr/sbin:/usr/bin"
Environment="PM2_HOME=/home/ec2-user/.pm2/pm2.pid"
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

[Service]
Type=systemd/system/pm2-ec2-user.service
CommandList
[+] 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] [+] Command successfully executed.
[PM2] Freezing a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup system
[ec2-user@ip-10-0-2-100 app-tier]$
```

29°C Partly sunny 15-09-2023

```

+----+-----+-----+
| id | amount | description |
+----+-----+-----+
| 1 | 400.00 | groceries |
| 2 | 400.00 | groceries |
+----+-----+-----+
2 rows in set (0.02 sec)

mysql> exit
Bye
[ec2-user@ip-10-0-2-100 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
     0          0      0 --:--:--:--:--:--:--:--:--:--:--:--:--:255k
100 14926  10 14926    0      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" ] && \source "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \source "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-2-100 bin]$ source ./bashrc
[ec2-user@ip-10-0-2-100 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...
100.0%
Extracting...
Checking checksum with sha256sum
Checksums matched!
Now using node v16.20.2 (npm v8.19.4)
Creating default alias: default > v16 (> v16.20.2)
[ec2-user@ip-10-0-2-100 bin]$ 

```

```

added 157 packages, and audited 150 packages in 0s
12 packages are looking for funding
  run 'npm fund' for details
Found 0 vulnerabilities
npm i
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-10-0-2-100 bin]$ cd ../
[ec2-user@ip-10-0-2-100 usr]$ pwd/usr
bash: pwd/usr: No such file or directory
[ec2-user@ip-10-0-2-100 usr]$ pwd /usr
/usr
[ec2-user@ip-10-0-2-100 usr]$ cd ..
[ec2-user@ip-10-0-2-100 ~]$ ls -rlt
ls: total 0
[ec2-user@ip-10-0-2-100 ~]$ ls is: command not found
[ec2-user@ip-10-0-2-100 ~]$ ls is: command not found
[ec2-user@ip-10-0-2-100 ~]$ ls -rlt
ls: total 0
drwxrwxr-x 2 ec2-user ec2-user 113 Sep 15 02:54 app-tier
[ec2-user@ip-10-0-2-100 ~]$ cd app-tier/

```

```

[ec2-user@ip-10-0-2-100 ~]$ ls -rlt
total 0
drwxrwxr-x 2 ec2-user ec2-user 113 Sep 15 02:54 app-tier
[ec2-user@ip-10-0-2-100 ~]$ cd app-tier/
[ec2-user@ip-10-0-2-100 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:

```

Session ID: root-03adfc1e4e6cde643      Instance ID: i-08da4677cdeaba389

Terminate

```

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] Starting /home/ec2-user/app-tier/index.js in fork_mode (1 instance)
[PM2] Done.



| <b>id</b> | <b>name</b> | <b>namespace</b> | <b>version</b> | <b>mode</b> | <b>pid</b> | <b>uptime</b> | <b>status</b> | <b>cpu</b> | <b>mem</b> | <b>user</b> | <b>watching</b> |          |
|-----------|-------------|------------------|----------------|-------------|------------|---------------|---------------|------------|------------|-------------|-----------------|----------|
| 0         | index       | default          | 1.0.0          | fork        | 70106      | 0s            | 0             | online     | 0%         | 26.4mb      | ec2-user        | disabled |


[ec2-user@ip-10-0-2-100 app-tier]$ 
```

29°C Partly sunny

Session ID: root-03adfc1e4e6cde643      Instance ID: i-08da4677cdeaba389

Terminate

```

type=forking
User=ec2-user
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/home/ec2-user/.nvm/versions/node/v16.20.2/bin:/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/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/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/cm2/bin/pm2 resurrect
ExecReload=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/cm2/bin/pm2 reload all
ExecStop=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/cm2/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-10-0-2-100 app-tier]$ 
```

29°C Partly sunny

Session ID: root-03adfc1e4e6cde643      Instance ID: i-08da4677cdeaba389

Terminate

```

Environment=PATH=/home/ec2-user/.nvm/versions/node/v16.20.2/bin:/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/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/cm2/bin/cm2 resurrect
ExecReload=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/cm2/bin/cm2 reload all
ExecStop=/home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/cm2/bin/cm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/cm2-ec2-user.service
Command list
[+] systemctl enable cm2-ec2-user...
[PM2] Writing init configuration in /etc/systemd/system/cm2-ec2-user.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable cm2-ec2-user...
Created symlink /etc/systemd/system/multi-user.target.wants/cm2-ec2-user.service → /etc/systemd/system/cm2-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-10-0-2-100 app-tier]$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/ec2-user/.pm2/dump.pm2
[ec2-user@ip-10-0-2-100 app-tier]$ curl http://localhost:4000/health
curl: (?) Failed to connect to localhost port 4000 after 0 ms: Couldn't connect to server
[ec2-user@ip-10-0-2-100 app-tier]$ 
```

29°C Partly sunny

## Step 12: Go to Ec2 instance

- In that Select the Instance we created and under Actions select image and template. Click Create image.

## Step 13: Create Target group

- Select Target type as Instance.
- 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

## Step14: Create Load balance

- 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
- Set the port to 80 and select The Target group will created.

## Step 15: Create Launch Template

- Set Instance Type as Free Tier
- Select The Private Instance SG in Security Group
- Select The IAM role we created

## Step 16 : Create Auto scaling

- Select our VPC
- Select the two private subnets AZ1,AZ2
- Select the existing load balancer we created
- Select the target groups we created
- Edit the Group Size and give value 2 in all three tables.

Screenshot of the AWS EC2 Instances page showing a list of running instances. One instance, 'myappserver1' (ID: i-08da4677cdeaba389), is selected. The 'Actions' dropdown menu is open, showing options like 'Connect', 'View details', 'Manage instance state', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'. A tooltip for 'Create Image' is visible.

Screenshot of the AWS Load Balancing Target Groups page. It shows a target group named 'Appliertargetgroup' with one target, 'Apptiertargetgroup', which is an instance with ID i-08da4677cdeaba389. The 'Actions' dropdown menu is open, showing options like 'Create target group', 'Edit', 'Delete', and 'Actions'.

Screenshot of the AWS Load Balancer creation wizard, Step 4: Configure Routing. It shows the configuration for an application load balancer. The protocol is set to 'HTTP' on port 4000. The 'Protocol version' section includes options for 'HTTP1', 'HTTP2', and 'gRPC'. The 'Health checks' section shows a configuration for 'HTTP' on path '/health'. The 'Next: Register Targets' button is visible at the bottom.

**Create launch template**

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

**Launch template name and description**

**Launch template name - required**  
Apptier-Launchtemplate  
Must be unique to this account. Max 128 chars. No spaces or special characters like & ", @, #.

**Template version description**  
Apptier-Launchtemplate  
Max 255 chars

**Auto Scaling guidance** Info  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

**Template tags**  
 Source template

**Summary**

**Software Image (AMI)**  
App tier  
ami-0b8ff2acc63922ba

**Virtual server type (instance type)**

**Firewall (security group)**

**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 month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

**Create launch template**

**Create launch template**

Search our full catalog including 1000s of application and OS images

**My AMIs**

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

**Amazon Machine Image (AMI)**

**App tier**  
ami-0b8ff2acc63922ba  
2023-09-15T03:18:01.000Z Virtualization: hvm ENA enabled: true Root device type: ebs boot mode: uefi-preferred

**Description**  
App tier

**Architecture** x86\_64   **AMI ID** ami-0b8ff2acc63922ba

**Summary**

**Software Image (AMI)**  
App tier  
ami-0b8ff2acc63922ba

**Virtual server type (instance type)**

**Firewall (security group)**

**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 month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

**Create launch template**

**Create Auto Scaling**

**Step 3 - optional**

**Configure advanced options**

**Step 4 - optional**

**Configure group size and scaling policies**

**Step 5 - optional**

**Add notifications**

**Step 6 - optional**

**Add tags**

**Step 7**

**Review**

**Load balancing** Info

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

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

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

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

**Attach to an existing load balancer**

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

**Existing load balancer target groups**

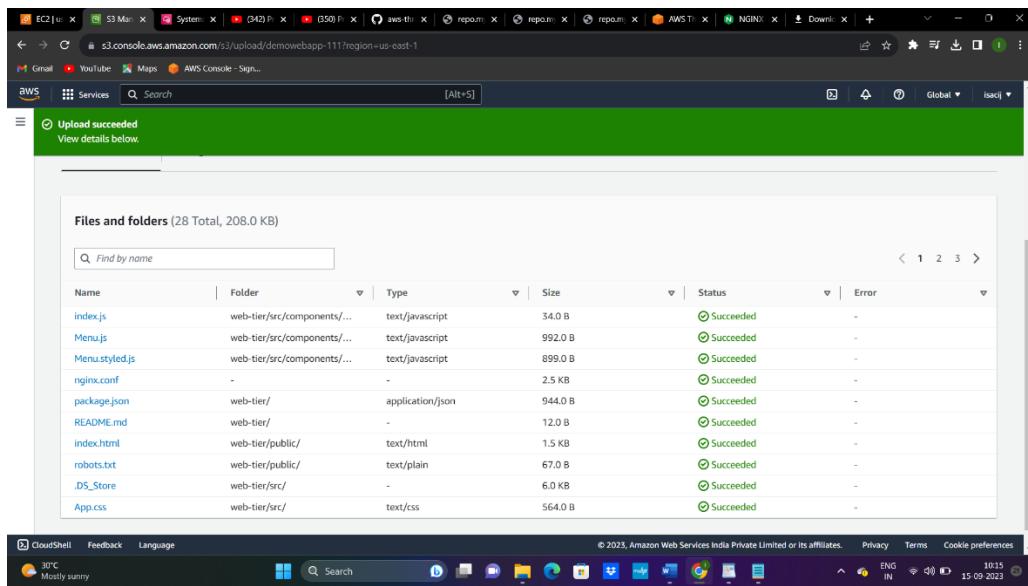
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

**Select target groups**

Appliertargetgroup | HTTP X Application Load Balancer: app-tier-internal

**CloudShell Feedback Language**

30°C Partly sunny



## Step 17: Create web instance

- Select our VPC .
- 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 ,which we created.
  
- Connect to terminal
  
- Use command sudo -su ec2-user
  
- Ping 8.8.8.8 to check the outside internet connectivity
  
- We now need to install all of the necessary components needed to run our front-end application. Again, start by installing NVM and node: curl -ohttps://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash source ~/.bashrc nvm .
- Install 16, nvm use 16

- we need to download our web tier code from our s3 bucket. Navigate to the web-layer folder and create the build folder for the react app so we can serve our code.
- Cd ~/web-tier , npm install , npm run build.
- NGINX can be used for different use cases like load balancing,

content caching etc, but we will be using it as a web server that we will configure to serve our application on port 80, as well as help direct our API calls to the internal load balancer. Use this command :sudo amazon-linux-extras install nginx1 -y.

- The Nginx configuration file with the following commands and list the files in the directory: 1.cd /etc/nginx ,2. ls.
- Replace the bucket name in the command below with the one you created for this workshop: 1. sudo rm nginx.conf ,2.sudo aws s3 cp s3:// demowebapp-111/nginx.conf .
- Then, restart Nginx with the following command : sudo service nginx restart
- To make sure Nginx has permission to access our files execute this command : chmod -R 755 /home/ec2-user
- And then to make sure the service starts on boot, run this command: sudo chkconfig nginx on.

**Launch an instance**

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags**

Name: demowebserver

**Application and OS Images (Amazon Machine Image)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

**Quick Start**

Recents | My AMIs | Quick Start

**Network settings**

VPC - required

vpc-0e5328fc1249f5b (awsdemo) 10.0.0.0/16

Subnet info

subnet-091a40a35c72e0a public-web-subnet Az-1 VPC vpc-0e5328fc1249f5b Owner: 821119645222 Availability Zone: us-east-1a IP addresses available: 250 CIDR: 10.0.0.0/24

Auto-assign public IP

Enable

Firewall (security group)

Create new security group | Select existing security group

Common security groups info

Select security groups | Compare security group rules

Advanced network configuration

**Summary**

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2... read more

Virtual server type (instance type): t2.micro

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 you launch

Cancel | Launch instance | Review commands

**CloudShell** Feedback Language

30°C Mostly sunny

CloudShell Feedback Language

30°C Mostly sunny

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

Gmail YouTube Maps AWS Console Sign...

CloudShell Feedback Language

30°C Mostly sunny

CloudShell Feedback Language

30°C Mostly sunny

us-east-1.console.aws.amazon.com/systems-manager/session-manager/j-i353aa0fb618ef7?region=us-east-1

Gmail YouTube Maps AWS Console Sign...

Session ID: root-0v66306eac8b80299 Instance ID: i-0353aa0fb618ef7

Terminate

```

sh-5.2$ sudo -su ec2-user
[ec2-user@ip-10-0-0-129 bin]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 64(48) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=0.644 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=0.752 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=0.743 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=0.765 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=52 time=0.699 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=52 time=0.720 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=52 time=0.775 ms
...
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6229ms
rtt min/avg/max/mdev = 0.644/0.726/0.775/0.043 ms
[ec2-user@ip-10-0-0-129 bin]$ curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ./bashrc
  % Total    % Received % Xferd  Average Speed   Time   Current
          Dload  Upload Total Spent   Left  Speed
100 14926  100 14926    0     0  145k  0:--:-- --:--:--:--:--:-- 145k
--> 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" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
[ec2-user@ip-10-0-0-129 bin]$ source ~/.bashrc
[ec2-user@ip-10-0-0-129 bin]$ nvm install 16

```

CloudShell Feedback Language

30°C Mostly sunny

```

Computing checksum with sha256sum
Checksum mode: 6.20.2 (ppm v0.19.4)
Creating default alias: default ( -> v16.20.2 )
[ec2-user@ip-10-0-0-129 bin]$ cp s3://demowebapp-111/web-tier/ web-tier --recursive
download: s3://demowebapp-111/web-tier/README.md to web-tier/README.md
download: s3://demowebapp-111/web-tier/src/components/DatabaseDemo/DatabaseDemo.js to web-tier/src/components/DatabaseDemo/DatabaseDemo.js
download: s3://demowebapp-111/web-tier/src/components/Home/Home.js to web-tier/src/components/Home/Home.js
download: s3://demowebapp-111/web-tier/src/App.css to web-tier/src/App.css
download: s3://demowebapp-111/web-tier/src/components/Menu/Menu.js to web-tier/src/components/Menu/Menu.js
download: s3://demowebapp-111/web-tier/package.json to web-tier/package.json
download: s3://demowebapp-111/web-tier/src/index.js to web-tier/src/index.js
download: s3://demowebapp-111/web-tier/src/components/Burger/Burger/index.js to web-tier/src/components/Burger/Burger/index.js
download: s3://demowebapp-111/web-tier/src/App.test.js to web-tier/src/App.test.js
download: s3://demowebapp-111/web-tier/src/App.js to web-tier/src/App.js
download: s3://demowebapp-111/web-tier/src/components/Menu/index.js to web-tier/src/components/Menu/index.js
download: s3://demowebapp-111/web-tier/src/components/Menu/Menu.styled.js to web-tier/src/components/Menu/Menu.styled.js
download: s3://demowebapp-111/web-tier/src/hooks.js to web-tier/src/hooks.js
download: s3://demowebapp-111/web-tier/src/reportWebVitals.js to web-tier/src/reportWebVitals.js
download: s3://demowebapp-111/web-tier/src/components/Burger/Burger.js to web-tier/src/components/Burger/Burger.js
download: s3://demowebapp-111/web-tier/src/index.css to web-tier/src/index.css
download: s3://demowebapp-111/web-tier/src/assets/3TierArch/index.html to web-tier/src/assets/3TierArch/index.html
download: s3://demowebapp-111/web-tier/src/global.js to web-tier/src/global.js
download: s3://demowebapp-111/web-tier/src/assets/3TierArch.png to web-tier/src/assets/3TierArch.png
download: s3://demowebapp-111/web-tier/src/theme.js to web-tier/src/theme.js
download: s3://demowebapp-111/web-tier/src/setupTests.js to web-tier/src/setupTests.js
download: s3://demowebapp-111/web-tier/public/robots.txt to web-tier/public/robots.txt
download: s3://demowebapp-111/web-tier/src/components/Burger/Burger.styled.js to web-tier/src/components/Burger/Burger.styled.js
download: s3://demowebapp-111/web-tier/src/index.js to web-tier/src/index.js
download: s3://demowebapp-111/web-tier/.DS_Store to web-tier/src/.DS_Store
download: s3://demowebapp-111/web-tier/src/components/.DS_Store to web-tier/src/components/.DS_Store
[ec2-user@ip-10-0-0-129 ~]$ 

```

```

total 24
-rw-rw-r--. 1 ec2-user ec2-user 944 Sep 15 04:44 package.json
-rw-rw-r--. 1 ec2-user ec2-user 12 Sep 15 04:44 README.md
drwxrwxr-x. 2 ec2-user ec2-user 42 Sep 15 04:59 public
drwxrwxr-x. 4 ec2-user ec2-user 16384 Sep 15 04:59 src
[ec2-user@ip-10-0-0-129 ~]$ npm audit
npm WARN deprecated core-js@1.4.2; Please use @jridgewell/source-map-codec instead
npm WARN deprecated rollup-plugin-terser@0.2.0; This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm WARN deprecated stable@0.1.8; Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated workbox-cacheable-response@6.6.0; workbox-background-sync@6.6.0
npm WARN deprecated w3c-hr-time@1.0.2; Your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated @babel/plugin-proposal-optimal-chaining@7.21.0; This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optimal-chaining 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-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-proposal-nullish-coalescing-operator 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 @babel/plugin-proposal-private-methods@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-private-methods instead.
npm WARN deprecated swgo@1.3.2; This SWGO version is no longer supported. Upgrade to v2.x.x.

added 1484 packages, and audited 1485 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

```

```

> aws-stier-web-layer@0.1.0 build
> react-scripts build

Creating an optimized production build...
One of your dependencies, babel-preset-react-app, is importing the "@babel/plugin-proposal-private-property-in-object" package without declaring it in its dependencies. This is currently working because "@babel/plugin-proposal-private-property-in-object" is already in your node_modules folder for unrelated reasons, but it may break at any time.

babel-preset-react-app is part of the create-react-app project, which is not maintained anymore. It is thus unlikely that this bug will ever be fixed. Add "@babel/plugin-proposal-private-property-in-object" to your dependencies to work around this error. This will make this message go away.

Compiled successfully.

File sizes after gzip:
 77.45 KB  build/static/js/main.9c9762a7.js
 1.79 KB   build/static/js/787.1f63e066.chunk.js
 493 B     build/static/css/main.b20b6ac4.css

The project was built assuming it is hosted at ./.
You can control this with the homepage field in your package.json.

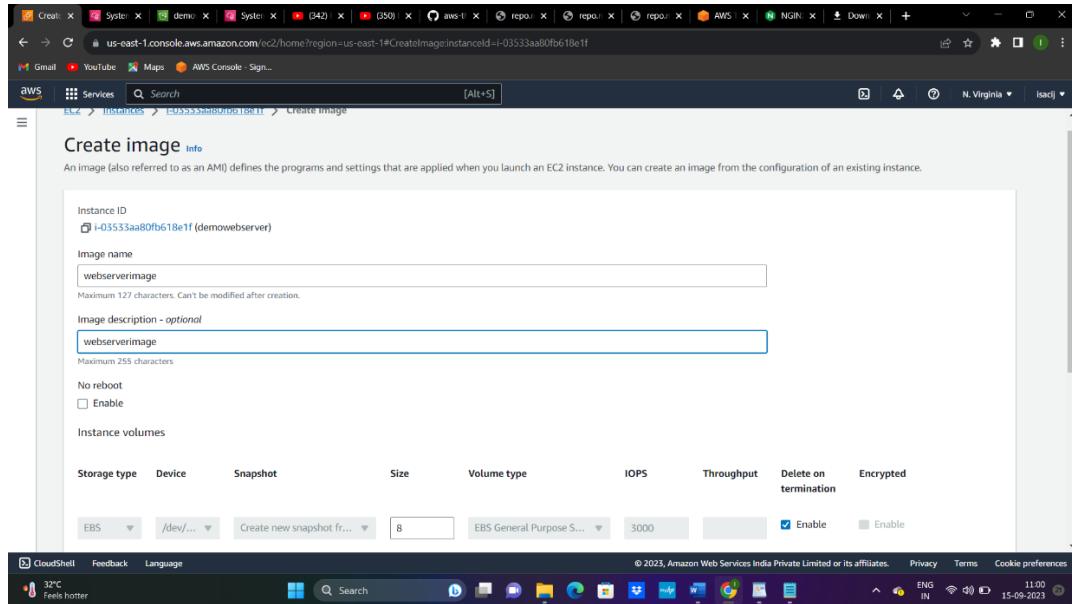
The build folder is ready to be deployed.

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

[ec2-user@ip-10-0-0-129 web-tier]$ 

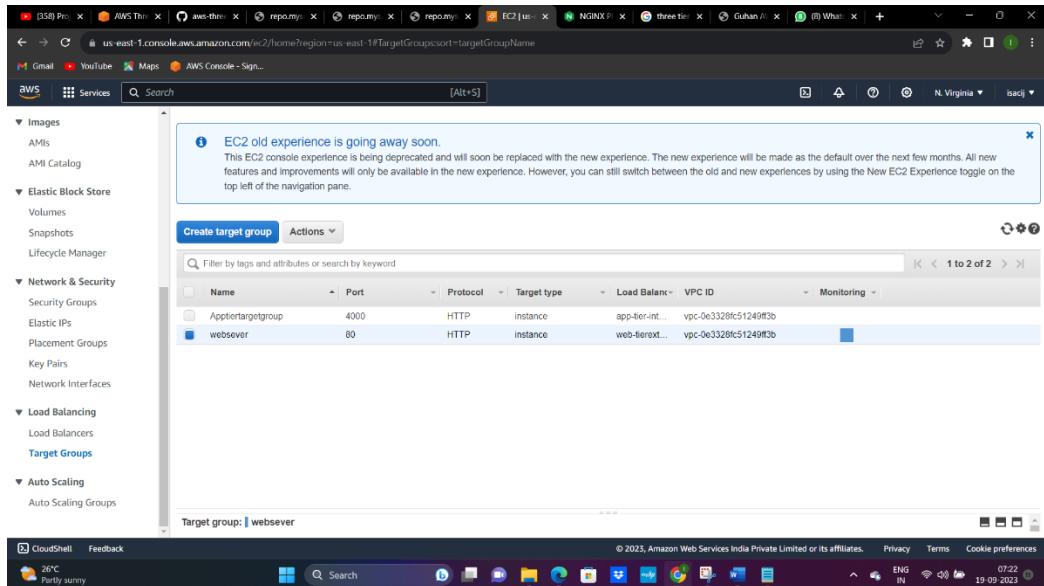
```

## Step 18 : Create image for AMI our Web instance



## Step 19: Create Target group

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



## Step 20: Create Load balancer

- Select the scheme to Internet Facing
- Select our VPC
- Select two public web subnets AZ1, AZ2
- Select security group which is Internet Facing External Load balancer
- Routing select port as 80 and forward to Web Target Group

The screenshot shows the AWS EC2 Load Balancers page. It displays two existing load balancers:

Name	DNS Name	State	VPC ID	Availability Zones	Type	Created
internal-internet	internal-internet.internall509	Active	vpc-0e3328fc51249f3b	us-east-1a, us-east-1b	application	September 15, 2023 at 11:04:53 AM UTC+5:30
anp-sec.internl	internal-sec.internl509	Active	vpc-0e3328fc51249f3b	us-east-1a, us-east-1b	application	September 15, 2023 at 11:04:53 AM UTC+5:30

Below the table, there is a "Create Load Balancer" button and a "Actions" dropdown menu.

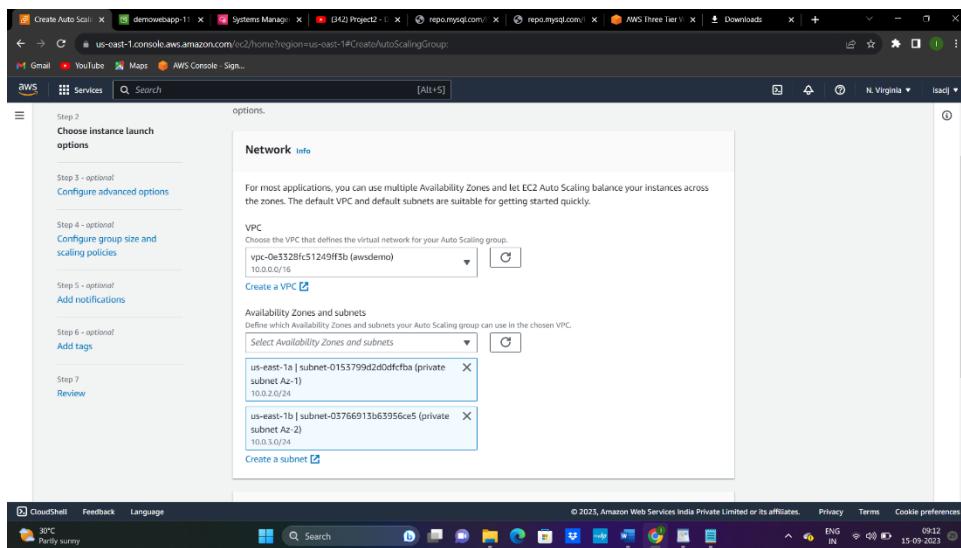
**Security** section details:

- Security groups: sg-016dd20a2d73894ca, Internetfacing-lb-sg
- External load balancer security groups

**Attributes** section is currently empty.

## Step 21: Create Launch template

- Select our Web instance AMI from MY AMI.
- Instance type as free tier.
- Select security group: Web tier SG.
- Select our IAM Role .



## Step 22: Create Auto scaling group

- Select our Web Launch Template.
- Select our VPC.
- Select the two public web subnets AZ1,AZ2 .
- Attach to existing load balancer we created for WEB.
- Select the WEB Target group.
- Edit the Group Size and give values 2 for all three tables.

The screenshot displays the AWS CloudShell interface with two browser tabs open:

- Top Tab:** Shows the EC2 Auto Scaling groups page. It lists two Auto Scaling groups:
 

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zone
webserverg	webservergLaunchtemp   Version Default	0	Updating capacity...	2	2	2	us-east-1a, ...
ApptierASG	Apptier-Lauchtemplate   Version Default	2	-	2	2	2	us-east-1a, ...

 A message at the bottom says "0 Auto Scaling groups selected".
- Bottom Tab:** Shows the EC2 Instances page. It lists six running instances:
 

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-0ec15e013f1e661f	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	-
-	i-056eb4d574743cc25	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
-	i-01a608378438fbe14	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
demowebserver	i-03553aa0fb618ef	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
myappserver1	i-08da5e4677cdeaba389	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	-
-	i-02da5e34f5be6ed44	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	-

 A modal window titled "Select an instance" is open, prompting the user to choose an instance from the list.

Step 23: And After this doing this go to load balancer and copy the DNS named as Web tier and Paste it in the browser.

