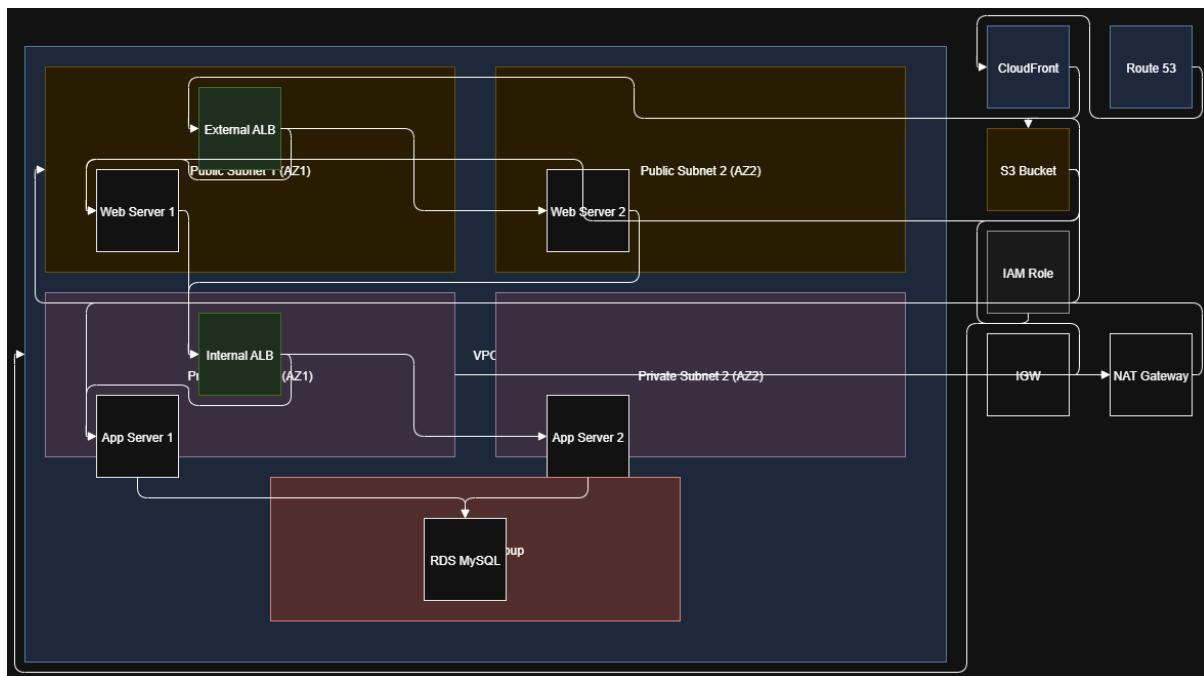


AWS 3-Tier Project (Webtier,Apptier,Dbtier)

Overview

I successfully designed and deployed a **highly available and secure 3-tier web application architecture** on AWS. This setup ensures scalability, reliability, and security by leveraging multiple AWS services and best practices for infrastructure design.

Architecture Diagrams



Service / Tool	Purpose
VPC (Virtual Private Cloud)	To create a dedicated, isolated network for the 3-tier architecture
Subnets	To logically separate public and private resources (Web / App / DB tiers)
Internet Gateway	Allows instances in public subnets to communicate with the internet
NAT Gateway	Allows private subnet instances to access the internet securely for updates
Route Tables	Controls routing of traffic between subnets and to/from the internet/NAT

Service / Tool	Purpose
Security Groups	Acts as a virtual firewall to control inbound/outbound traffic per tier
EC2 Instances	Hosts the Web Server and Application Server workloads
Application Load Balancer (External)	Distributes internet traffic to Web Tier (public-facing)
Application Load Balancer (Internal)	Distributes traffic from Web Tier to App Tier (private)
Auto Scaling Group	Automatically scales EC2 instances based on demand (both Web & App tiers)
RDS (MySQL)	Managed relational database in the Database Tier
IAM Role + S3 ReadOnly Policy	Allows EC2 instances to securely pull code from S3 buckets
Elastic IP	Provides a fixed public IP for NAT Gateway (used in Private subnets)
CloudFront (CDN)	Caches and delivers web content with low latency and SSL termination
ACM (AWS Certificate Manager)	Provides the SSL certificate used by CloudFront for HTTPS access
Route 53 (DNS Service)	Manages domain names and routes user traffic to CloudFront / Load Balancer with DNS mapping

→Clone the code from Repo

Download the code from [this repository](https://catalog.us-east-1.prod.workshops.aws/workshops/85cd2bb2-7f79-4e96-bdee-8078e469752a/en-US/part0/code) into your local environment by running the command below. If you don't have git installed, you can just download the zip. Save it somewhere you can easily access.

From <<https://catalog.us-east-1.prod.workshops.aws/workshops/85cd2bb2-7f79-4e96-bdee-8078e469752a/en-US/part0/code>>

Create Folder Named : AWS-Project

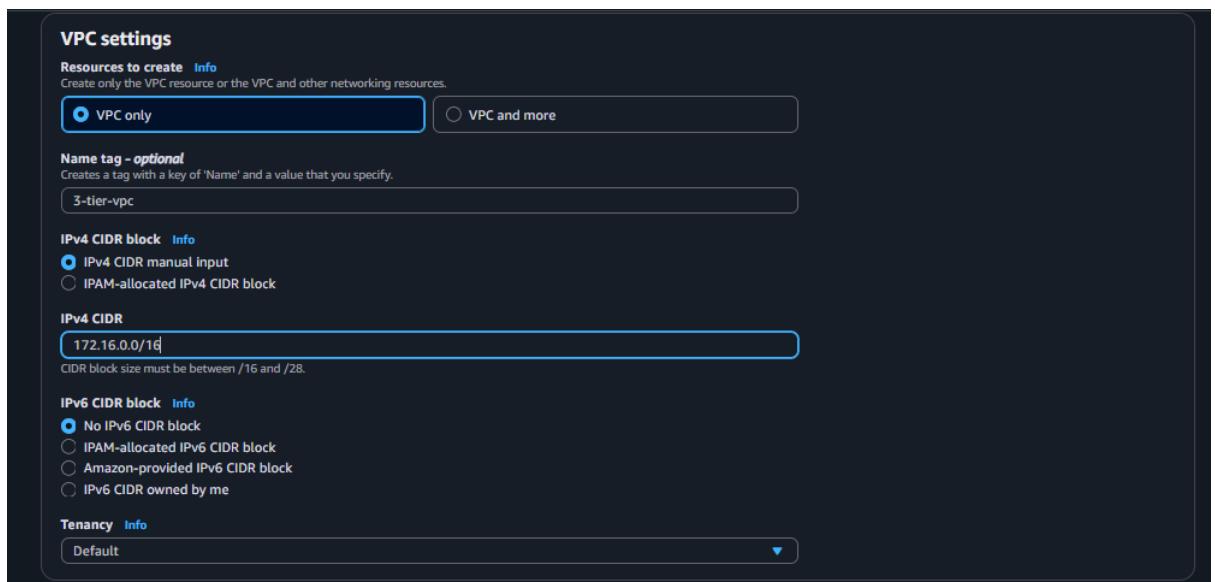
Open the folder and Open gitbash or command prompt for this location (if you are not install gitbash install : [<https://git-scm.com/downloads>](https://git-scm.com/downloads))

And clone the code from remote by ref below url :

git clone <https://github.com/aws-samples/aws-three-tier-web-architecture-workshop.git>

Step 1 – Create a VPC (Virtual Private Cloud)

1. In the AWS Console, search for **VPC** and open the VPC Dashboard.
2. Select **Create VPC** and choose *VPC only* option.
3. Provide a name (e.g., “3-tier-vpc”) and CIDR block (e.g., 172.16.0.0/16), then click **Create VPC**.



Step 2 – Create Subnets

1. In the VPC Dashboard, select **Subnets** and choose **Create subnet**.
2. Add 2 public subnets (e.g., 172.16.1.0/24 and 172.16.2.0/24) and 2 private subnets (e.g., 172.16.3.0/24 and 172.16.4.0/24).
3. Ensure the subnets are spread across at least 2 Availability Zones.
 - Select your custom VPPC

Create subnet Info

VPC
Create subnets in this VPC.

Select a VPC

(default)

Select a VPC first to create new subnets.

Add new subnet

Cancel Create subnet

Subnet-1→Public-subnet

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

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

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

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

IPv4 subnet CIDR block
 256 IPs
< > ^ ↓

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="public-subnet-01"/> <small>X Remove</small>

Add new tag

Subnet-2→Public-subnet

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

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

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

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

IPv4 subnet CIDR block
 256 IPs
< > ^ ↓

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="public-subnet-02"/> <small>X Remove</small>

Add new tag

Subnet-3→Private-subnet

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

The name can be up to 256 characters long.

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

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

IPv4 subnet CIDR block
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private-subnet-01"/>

You can add 49 more tags.

Subnet-4→Private-subnet

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

The name can be up to 256 characters long.

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

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

IPv4 subnet CIDR block
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private-subnet-02"/>

You can add 49 more tags.

Subnets (4) Info

Last updated 2 minutes ago

<input type="checkbox"/>	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	public-subnet-01	subnet-021b4ea9190085709	<input checked="" type="checkbox"/> Available	vpc-07a4339e7f23981e0 3-tie...	<input type="radio"/> Off	172.16.1.0/24	-
<input type="checkbox"/>	public-subnet-02	subnet-0a1b6fe5215ace94	<input checked="" type="checkbox"/> Available	vpc-07a4339e7f23981e0 3-tie...	<input type="radio"/> Off	172.16.2.0/24	-
<input type="checkbox"/>	private-subnet-02	subnet-0ea47676f5e41442	<input checked="" type="checkbox"/> Available	vpc-07a4339e7f23981e0 3-tie...	<input type="radio"/> Off	172.16.4.0/24	-
<input type="checkbox"/>	private-subnet-01	subnet-05d43d5fb9e171eb	<input checked="" type="checkbox"/> Available	vpc-07a4339e7f23981e0 3-tie...	<input type="radio"/> Off	172.16.3.0/24	-

Step 3 – Configure Internet Gateway

- In the VPC Dashboard, select **Internet Gateways** and create a new Internet Gateway.

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.
3-tier-internet-gateway

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

Key	Value - optional
Q Name	Q 3-tier-internet-gateway

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

2. Attach the Internet Gateway to your VPC.

- In the action button select attach to VPC

igw-09444976c598ac676 / 3-tier-internet-gateway

Details Info

Internet gateway ID igw-09444976c598ac676	State Detached	VPC ID -	Owner 941098798453
--	-------------------	-------------	-----------------------

Tags

Key	Value
Name	3-tier-internet-gateway

Actions ▾

- Attach to VPC
- Detach from VPC
- Manage tags
- Delete

Manage tags

3. Select your custom VPC then click on “Attach internet gateway”

Attach to VPC (igw-09444976c598ac676) Info

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

Q Select a VPC
vpc-07a4339ef23981e0 - 3-tier-vpc

Cancel Attach internet gateway

Step 4 – Configure Route Tables

1. In Route Tables, create one route table for public subnets and another for private subnets.

Public-Route-Table:

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.
public-route-table

VPC
The VPC to use for this route table.
vpc-07a4339e7f23981e0 (3-tier-vpc)

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="Name"/>	<input type="text" value="public-route-table"/> X Remove

Add new tag You can add 49 more tags.

Cancel Create route table

Private-Route-Table:

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.
private-route-table

VPC
The VPC to use for this route table.
vpc-07a4339e7f23981e0 (3-tier-vpc)

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="Name"/>	<input type="text" value="private-route-table"/> X Remove

Add new tag You can add 49 more tags.

Cancel Create route table

- Add a route (0.0.0.0/0) to the public route table and target the Internet Gateway. Click on “Edit routes”

rtb-090f3d830a3880cac / public-route-table

Actions ▾

Details Info

Route table ID <input type="text" value="rtb-090f3d830a3880cac"/>	Main <input type="checkbox"/> No	Explicit subnet associations -	Edge associations -
VPC <input type="text" value="vpc-07a4339e7f23981e0 3-tier-vpc"/>	Owner ID <input type="text" value="941098798453"/>		

Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Destination		Target	Status	Propagated	Route Origin
172.16.0.0/16	local	Active	No	Create Route Table	

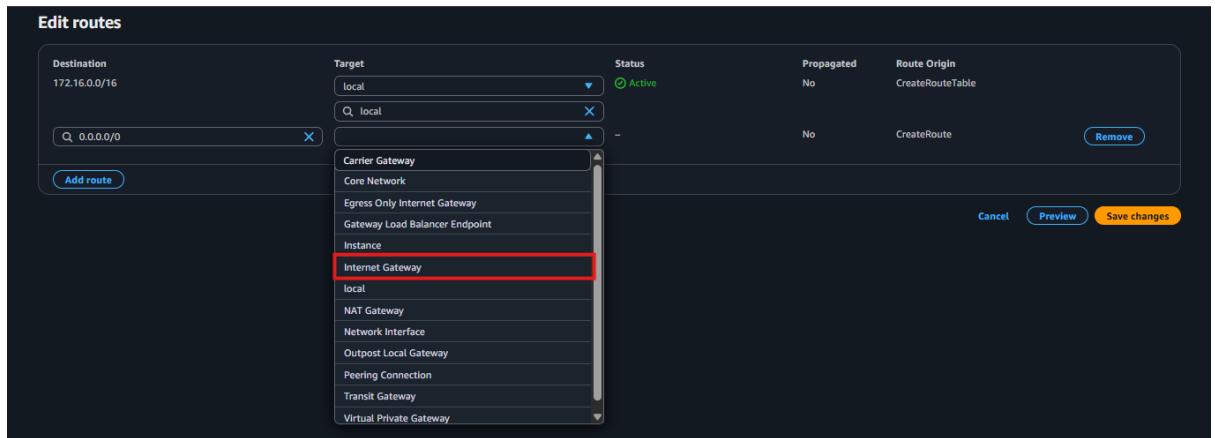
Both Edit routes 1 Create Route Table

- Click on “Add route” button

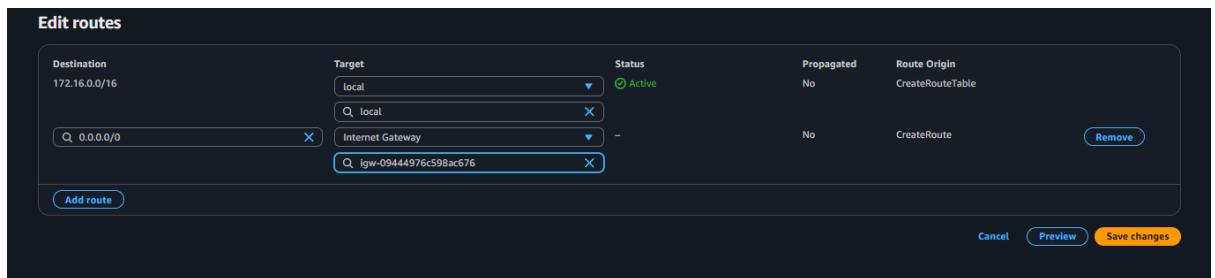
Edit routes

Destination	Target	Status	Propagated	Route Origin
172.16.0.0/16	<input type="text" value="local"/> X	Active	No	CreateRouteTable

Add route Cancel Preview Save changes

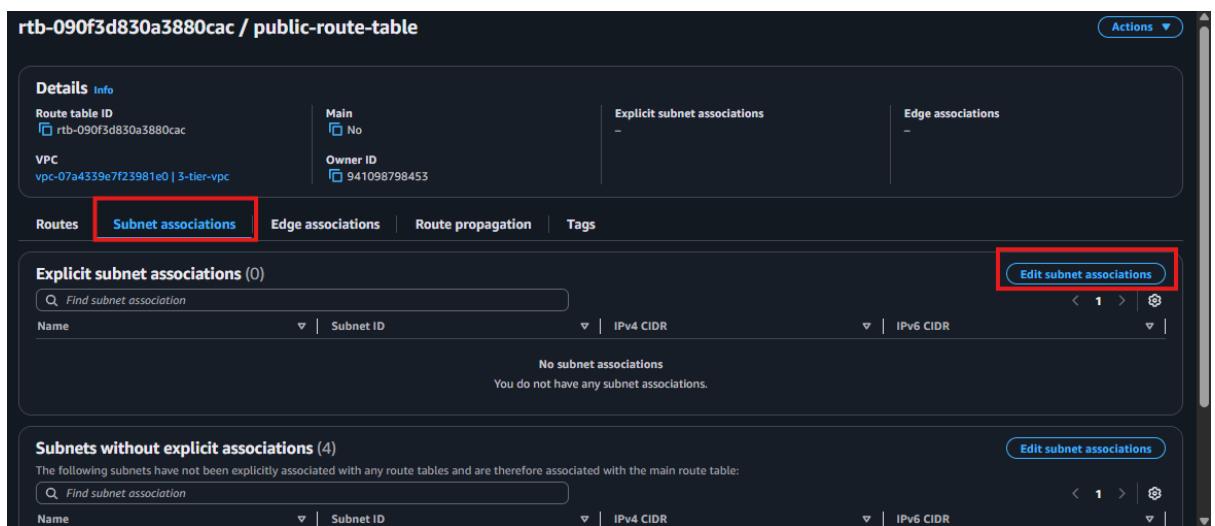


- Select your internet gateway and “Save changes”



3. Associate public subnets with the public route table.

- Click “Subnet associations” then click “Edit subnet associations”



- Select the public subnets and click on “Save associations”

Available subnets (2/4)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> public-subnet-01	subnet-021b4ea9190085709	172.16.1.0/24	-	Main (rtb-038ed063c3ca28687)
<input checked="" type="checkbox"/> public-subnet-02	subnet-0a1bc6fe3215ace94	172.16.2.0/24	-	Main (rtb-038ed063c3ca28687)
<input type="checkbox"/> private-subnet-02	subnet-0e8a476f5e41442	172.16.4.0/24	-	Main (rtb-038ed063c3ca28687)
<input type="checkbox"/> private-subnet-01	subnet-05d43d3fb9e171eb	172.16.3.0/24	-	Main (rtb-038ed063c3ca28687)

Selected subnets

subnet-021b4ea9190085709 / public-subnet-01 X subnet-0a1bc6fe3215ace94 / public-subnet-02 X

Cancel **Save associations**

4. Associate private subnets with the private route table.

- Click “Subnet associations” then click “Edit subnet associations”

rtb-038ad7f63223be8db / private-route-table

Details **Info**

Route table ID rtb-038ad7f63223be8db	Main <input type="checkbox"/> No	Explicit subnet associations -	Edge associations -
VPC vpc-07a4339e7f23981e0 3-tier-vpc	Owner ID 941098798453		

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

Explicit subnet associations (0)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations You do not have any subnet associations.			

- Select the private subnets and click on “Save associations”

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/> public-subnet-01	subnet-021b4ea9190085709	172.16.1.0/24	-	rtb-090f3d30a3880cac / public-route...
<input type="checkbox"/> public-subnet-02	subnet-0a1bc6fe3215ace94	172.16.2.0/24	-	rtb-090f3d30a3880cac / public-route...
<input checked="" type="checkbox"/> private-subnet-02	subnet-0e8a476f5e41442	172.16.4.0/24	-	Main (rtb-038ed063c3ca28687)
<input checked="" type="checkbox"/> private-subnet-01	subnet-05d43d3fb9e171eb	172.16.3.0/24	-	Main (rtb-038ed063c3ca28687)

Selected subnets

subnet-0e8a476f5e41442 / private-subnet-02 X subnet-05d43d3fb9e171eb / private-subnet-01 X

Cancel **Save associations**

Step 4a – Create NAT Gateway

1. In the VPC Dashboard, select **NAT Gateways** and click **Create NAT Gateway**.
2. Select one of the public subnets and associate an Elastic IP address.

NAT gateway settings

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

Subnet
Select a subnet in which to create the NAT gateway.

 subnet-087e5364e7c8b41c62 (default-subnet)
 us-east-1a
subnet-021b4ea9190085709 (public-subnet-01)
 us-east-1a
 subnet-031bc6fe3e215ace94 (public-subnet-02)
 us-east-1b
 subnet-0e8a47676f5e41442 (private-subnet-02)
 us-east-1b
 subnet-05d43d5fb9e171eb (private-subnet-01)
 us-east-1a

- Then click on “Allocate Elastic IP”

NAT gateway settings

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

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.
 Public
 Private

Elastic IP allocation ID - Info
Assign an Elastic IP address to the NAT gateway.

- And click the “Save changes”

3. Create the NAT Gateway and wait until it becomes available.

NAT gateways (1) - Info

Name	NAT gateway ID	Connectivity...	State	Primary public IP...	Primary private I...	Primary network...
nat-gateway	nat-017505879d455384f	Public	Available	3.213.93.12	172.16.1.125	eni-057e06428a5c

4. In the private route table, add a route (0.0.0.0/0) and target the new NAT Gateway.

rtb-038ad7f63223be8db / private-route-table

Details Info

Route table ID rtb-038ad7f63223be8db	Main <input type="checkbox"/> No	Explicit subnet associations -	Edge associations -
VPC vpc-07a4339e7f23981e0 3-tier-vpc	Owner ID 941098798453		

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

Routes (1)

Destination	Target	Status	Propagated	Route Origin
172.16.0.0/16	local	Active	No	Create Route Table

Edit routes

- Click on “Add route” and select “NAT Gateway” then in the drop down select the NAT we have just created. And enter the “Save changes”

Edit routes

Destination 172.16.0.0/16	Target <input type="text" value="local"/> <input type="button" value="Q local"/>	Status Active	Propagated No	Route Origin CreateRouteTable
------------------------------	---	---	------------------	----------------------------------

Add route

Cancel **Preview** **Save changes**

Edit routes

Destination 172.16.0.0/16	Target <input type="text" value="local"/> <input type="button" value="Q 0.0.0.0"/>	Status Active	Propagated No	Route Origin CreateRouteTable
------------------------------	---	---	------------------	----------------------------------

Add route

Carrier Gateway **Core Network** **Egress Only Internet Gateway** **Gateway Load Balancer Endpoint** **Instance** **Internet Gateway** **local** **NAT Gateway** **Network Interface** **Outpost Local Gateway** **Peering Connection** **Transit Gateway** **Virtual Private Gateway**

Remove **Cancel** **Preview** **Save changes**

The screenshot shows the 'Edit routes' page for a VPC. A new route is being added to a NAT Gateway. The destination is 172.16.0.0/16, and the target is set to 'NAT Gateway' with the specific entry 'nat-017505879d455384f' selected. The route is labeled 'nat-017505879d455384f (nat-gateway)'.

- Finally your VPC → **Resource map** look like this we are done with “Network part”.



Step 5 – Create Security Groups

Create the following six custom security groups:

- Jump Server SG – allows SSH (port 22) from all public IP addresses.

The screenshot shows the 'Create security group' interface. The 'Basic details' section includes:

- Security group name:** Jump-server (highlighted with a red box).
- Description:** Allows SSH access for remote the server (highlighted with a red box).
- VPC:** vpc-07a4339e7f23981e0 (3-tier-vpc) (highlighted with a red box).

 The 'Inbound rules' section shows a single rule:

- Type:** SSH (highlighted with a red box).
- Protocol:** TCP.
- Port range:** 22.
- Source:** Anywhere (highlighted with a red box).
- Description - optional:** for all public access (highlighted with a red box).

 A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." (highlighted with a red box).

- External Load Balancer SG – allows HTTP (port 80) from all public IP addresses.

Create security group Info

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

Basic details

Security group name Info
external-load-balancer
Name cannot be edited after creation.

Description Info
allows public traffic to web-tier

VPC Info
vpc-07a4339e7f23981e0 (3-tier-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere... <input type="text" value="0.0.0.0/0"/>	<input type="text" value="0.0.0.0"/> <small>X</small>

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

- Web Server SG – allows HTTP (port 80) from the External Load Balancer SG and SSH (port 22) from the Jump Server SG.

Create security group Info

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

Basic details

Security group name Info
web-server
Name cannot be edited after creation.

Description Info
allows external load-balancer traffic

VPC Info
vpc-07a4339e7f23981e0 (3-tier-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Custom <input type="text" value="sg-0255a3f2a977ddb94"/>	<input type="text" value="sg-0255a3f2a977ddb94"/> <small>X</small> <small>Delete</small>
SSH	TCP	22	Custom <input type="text" value="sg-01ce4f433eea46ca0"/>	<input type="text" value="sg-01ce4f433eea46ca0"/> <small>X</small> <small>Delete</small>

Add rule

- Internal Load Balancer SG – allows HTTP (port 80) from the Web Server SG.

Create security group Info

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

Basic details

Security group name Info
internal-load-balancer

Name cannot be edited after creation.

Description Info
allows traffic from web-server

VPC Info
vpc-07a4339e7f23981e0 (3-tier-vpc)

Inbound rules Info

Type	Protocol	Port range	Source
HTTP	TCP	80	Custom

[Add rule](#)

Security Groups

Use: "sg-010e530c8c367d0d5"

CIDR blocks

Prefix lists

Description - optional

Delete

Q sg-010e530c8c367d0d5 X

sg-010e530c8c367d0d5 X

5. App Server SG – allows port 4000 from the Internal Load Balancer SG. And Allow port 22 from the Jump-server SG for remote access from jump-server to app-server

Create security group Info

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

Basic details

Security group name Info
app-server

Name cannot be edited after creation.

Description Info
allows traffic flow from internal load-balancer

VPC Info
vpc-07a4339e7f23981e0 (3-tier-vpc)

Inbound rules Info

Type	Protocol	Port range	Source
Custom TCP	TCP	4000	Custom

[Add rule](#)

Security Groups

internal-load-balancer | sg-061b3bbf2e9feef7d

internal-load-balancer

Use: "sg-061b3bbf2e9feef7d"

CIDR blocks

Prefix lists

Description - optional

Delete

Q sg-061b3bbf2e9feef7d X

sg-061b3bbf2e9feef7d X

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID
sgr-082fd4bb48864563d

Type Info
Custom TCP

Protocol Info
TCP

Port range Info
4000

Source Info
Custom

Description - optional Info
Q sg-061b3bbf2e9feef7d X

Delete

-

Type Info
SSH

Protocol Info
TCP

Port range Info
22

Source Info
Custom

Description - optional Info
Q sg-01ce4f433eea46ca0 X

Delete

Security Groups

Jump-server | sg-01ce4f433eea46ca0

Jump-server

Use: "sg-01ce4f433eea46ca0"

CIDR blocks

Prefix lists

Description - optional

Delete

Cancel

Preview changes

Save rules

6. Database SG – allows MySQL (port 3306) from the App Server SG

Create security group Info

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

Basic details			
Security group name <small>Info</small>			
database			
Name cannot be edited after creation.			
Description <small>Info</small>			
allows db to app-server			
VPC Info			
vpc-07a4339e7f23981e0 (3-tier-vpc)			
Inbound rules <small>Info</small>			
Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>
MySQL/Aurora	TCP	3306	Custom
Security Groups app-server sg-034f16022939e2a29 app-server			
Prefix lists Q_ sg-034f16022939e2a29 X sg-034f16022939e2a29 X			
Description - optional <small>Info</small>			
Delete			
Add rule			

Step 6 – Create RDS Subnet Group

1. In the RDS dashboard, select Subnet groups and click Create DB subnet group.
2. Provide a name and description.

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details	
Name	
database-1-subnet-group	
You won't be able to modify the name after your subnet group has been created.	
Description	
database-1-subnet-group	
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.	
3-tier-vpc (vpc-07a4339e7f23981e0) 4 Subnets, 2 Availability Zones	
3-tier-vpc (vpc-07a4339e7f23981e0) ✓ 4 Subnets, 2 Availability Zones	
Default VPC (vpc-03994dc57c3259ee) 1 Subnets, 1 Availability Zones	
Availability zones	
Choose the Availability Zones that include the subnets you want to add.	

3. Select your VPC and choose the AZ (Availability Zone) where the private subnets are created. Then add the private subnets that you created earlier. Finally click on “Create”.

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.
3-tier-vpc (vpc-07a4339e7f23981e0) 4 Subnets, 2 Availability Zones
3-tier-vpc (vpc-07a4339e7f23981e0) ✓ 4 Subnets, 2 Availability Zones
Default VPC (vpc-03994dc57c3259ee) 1 Subnets, 1 Availability Zones

The screenshot shows the 'Add subnets' step in the AWS Subnet creation wizard. In the 'Availability Zones' section, 'us-east-1a' and 'us-east-1b' are selected and highlighted with a red box. In the 'Subnets' section, 'private-subnet-01' and 'private-subnet-02' are selected and highlighted with a red box. The 'Create' button is highlighted with a red box at the bottom right.

Step 6a – Configure RDS Database

1. In the AWS Console, search for **RDS** and select **Create Database**.
2. Choose Standard Create and select the **Free Tier** options.

The screenshot shows the 'Create database' configuration page. The 'Choose a database creation method' section has 'Standard create' selected and highlighted with a red box. The 'Easy create' option is also visible but not selected.

3. Choose **MySQL** as the engine type

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible)	<input type="radio"/> Aurora (PostgreSQL Compatible)	<input checked="" type="radio"/> MySQL
<input type="radio"/> PostgreSQL	<input type="radio"/> MariaDB	<input type="radio"/> Oracle
<input type="radio"/> Microsoft SQL Server	<input type="radio"/> IBM Db2	IBM Db2

Edition [Info](#)

MySQL Community

Engine version [Info](#)

View the engine versions that support the following database features.

Engine version

MySQL 8.0.42

Enable RDS Extended Support [Info](#)

Amazon RDS Extended Support is a paid offering [Learn more](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

- In the “Templates” chose free tier.

Templates

Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.	<input checked="" type="radio"/> Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info
--	--	--

Availability and durability

Deployment options [Info](#)

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

<input type="radio"/> Multi-AZ DB cluster deployment (3 instances) Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides: <ul style="list-style-type: none">99.95% uptimeRedundancy across Availability ZonesImproved query capacityReduced write latency	<input type="radio"/> Multi-AZ DB instance deployment (2 instances) Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides: <ul style="list-style-type: none">99.95% uptimeRedundancy across Availability Zones	<input checked="" type="radio"/> Single-AZ DB instance deployment (1 instance) Creates a single DB instance without standby instances. This setup provides: <ul style="list-style-type: none">99.5% uptimeNo data redundancy
--	--	--

- configure DB name, username, and password.

Settings

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

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

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

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

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

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

Master password [Info](#)

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

Confirm master password [Info](#)

- Choose custom VPC and select the DB subnet group that we created early.

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.

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 instance.

3-tier-vpc (vpc-07a4339e7f23981e0)
4 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 instance can use in the VPC that you selected.

database-1-subnet-group
2 Subnets, 2 Availability Zones

- Choose the security group for the database allowing access only from the application tier security group. and disable public access, choose the AZ (availability zone).

Public access [Info](#)
 Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

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 the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

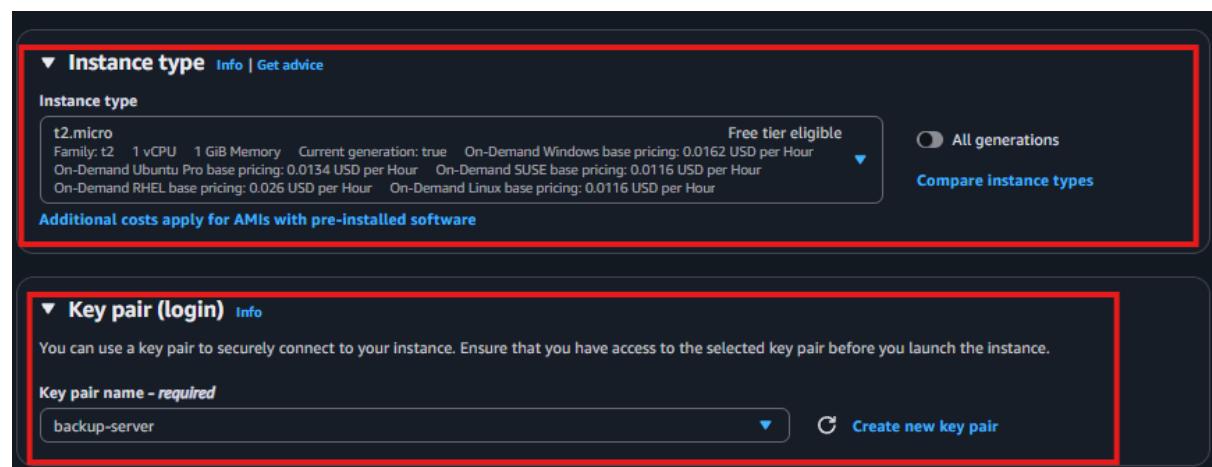
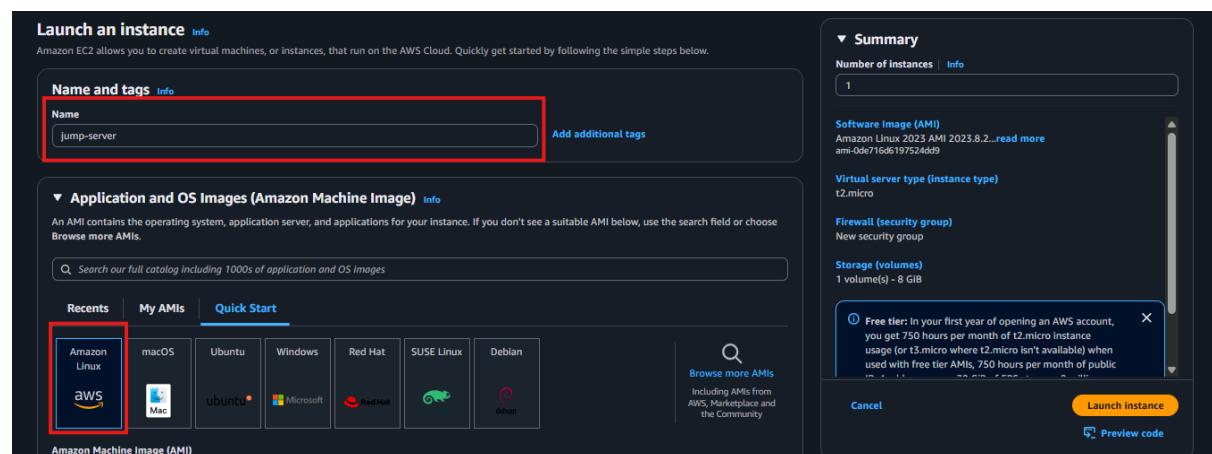
Existing VPC security groups

Availability Zone [Info](#)
us-east-1a

- Then Create database.

Step 7 – Launch Jump Server (Bastion Host)

1. Navigate to EC2 and click Launch Instance.
2. Choose an Amazon Linux 2 AMI and t2.micro instance type.



3. Select a public subnet and assign the Jump Server SG security groups

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-07a4339e7f23981e0 (3-tier-vpc)
172.16.0.0/16

Subnet | [Info](#)

subnet-021b4ea9190085709 public-subnet-01
VPC: vpc-07a4339e7f23981e0 Owner: 941098798453 Availability Zone: us-east-1a (use1-az4)
Zone type: Availability Zone IP addresses available: 250 CIDR: 172.16.1.0/24

Create new subnet [\[?\]](#)

Auto-assign public IP | [Info](#)

Enable

Additional charges apply when outside of [free tier allowance](#)

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

Jump-server sg-01ce4f433eea46ca0 X
VPC: vpc-07a4339e7f23981e0

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

4. Launch the instance and note the public IP address (used for SSH to other private servers).

Step 8 – Launch EC2 Instances (Application Tier)

1. Repeat the steps for launching new EC2 instances, but select the private subnets. And choose the security group that we have created early.

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-07a4339e7f23981e0 (3-tier-vpc)
172.16.0.0/16

Subnet | [Info](#)

subnet-05d43d3fb9e171eb private-subnet-01
VPC: vpc-07a4339e7f23981e0 Owner: 941098798453 Availability Zone: us-east-1a (use1-az4)
Zone type: Availability Zone IP addresses available: 250 CIDR: 172.16.3.0/24

Create new subnet [\[?\]](#)

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

app-server sg-034f16022939e2a29 X
VPC: vpc-07a4339e7f23981e0

Compare security group rules

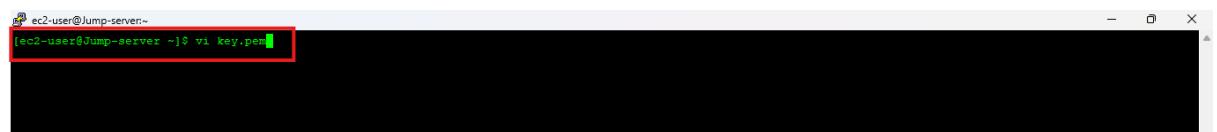
Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

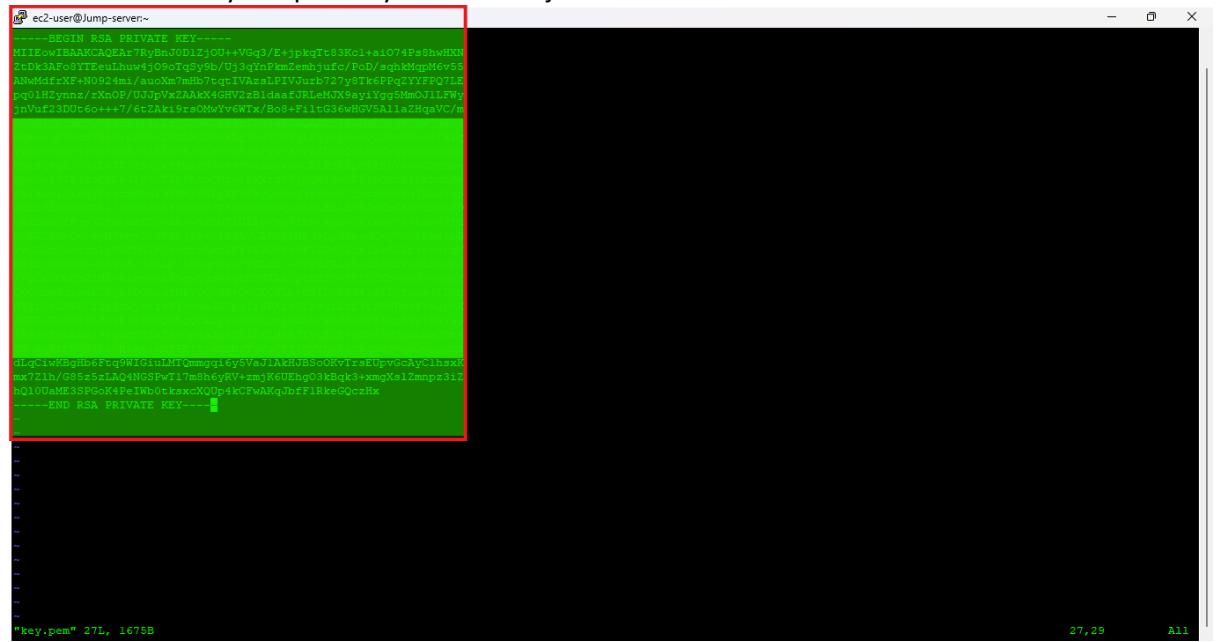
Step 9 - Open the Jump server and take ssh to configure App server and database.



- Create a file using your custom key name (like key.pem)



- Paste your .pem key into the file just created.



- Then change permission for the key file in the Jump-server
- sudo chmod 400 key.pem

```
[ec2-user@ip-172-16-3-13:~]
[ec2-user@Jump-server ~]$ sudo chmod 400 key.pem
[ec2-user@Jump-server ~]$ ssh -i key.pem ec2-user@172.16.3.13
      #
      _###_
      Amazon Linux 2023
      _###|
      \#/
      V~' '-->
      /
      /_\
      /m/
[ec2-user@ip-172-16-3-13 ~]$
```

- Then take remote of your app-server using ssh
- Bash: **ssh -i key.pem username@private IP**

```
[ec2-user@Jump-server:~]
[ec2-user@Jump-server ~]$ ssh -i key.pem ec2-user@172.16.3.13
```

```
[ec2-user@ip-172-16-3-13:~]
[ec2-user@Jump-server ~]$ sudo chmod 400 key.pem
[ec2-user@Jump-server ~]$ ssh -i key.pem ec2-user@172.16.3.13
      #
      _###_
      Amazon Linux 2023
      _###|
      \#/
      V~' '-->
      /
      /_\
      /m/
[ec2-user@ip-172-16-3-13 ~]$
```

At App-Server: Install MySQL Agent for running RdS-DB Engine

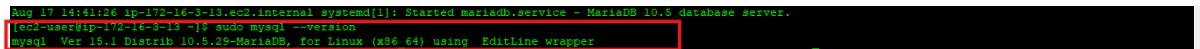
Commands:

Install MySQL Client :

```
yum list mariadb*
sudo yum install mariadb105-server.x86_64
sudo systemctl enable mariadb
sudo systemctl start mariadb
sudo systemctl status mariadb.service
```

```
sudo mysql --version
```

- After executed all the command the you will see the version of your mysql client

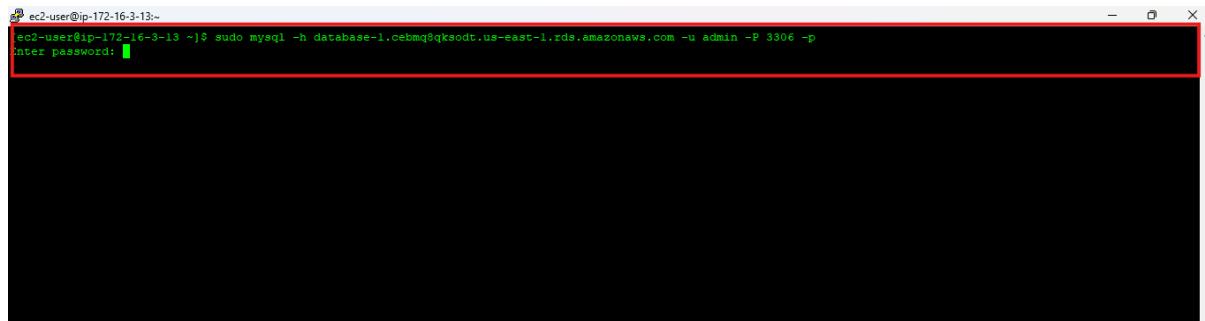


```
[Aug 17 14:41:26 ip-172-16-3-13.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.  
[ec2-user@ip-172-16-3-13 ~]$ sudo mysql --version  
mysql Ver 15.1 Distrib 10.5.29-MariaDB, for Linux (x86_64) using EditLine wrapper
```

Login to your MySql_DB Engine

Command :

```
mysql -h <Host_Name> -P 3306 -u <User_Name> -p then enter your password
```



```
[ec2-user@ip-172-16-3-13 ~]$ sudo mysql -h database-1.cebmq8qksdt.us-east-1.rds.amazonaws.com -u admin -P 3306 -p  
Enter password: [REDACTED]
```

For Ex

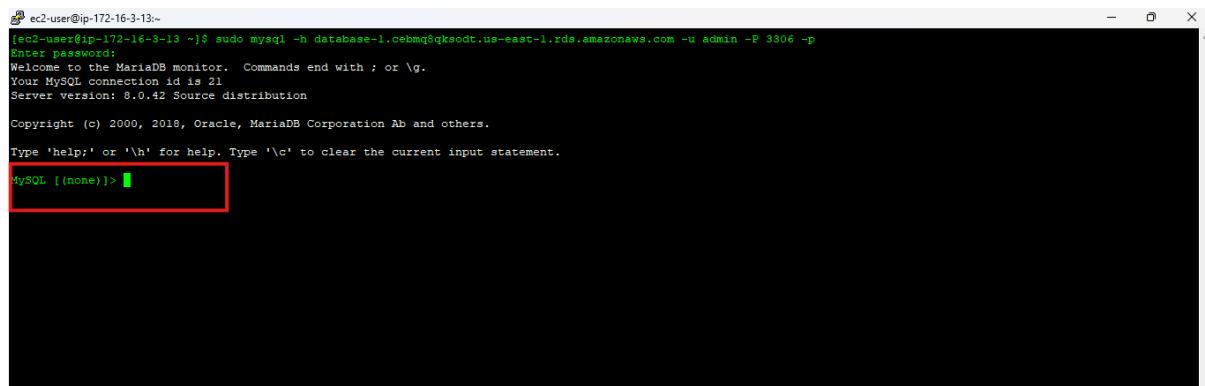
Hostname : database-1.cxz1od1wh1ta.us-east-1.rds.amazonaws.com

Username:admin

Passwrdf: Admin123

The command is : mysql -h database-1.cxz1od1wh1ta.us-east-1.rds.amazonaws.com -P 3306 -u admin -p

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



```
[ec2-user@ip-172-16-3-13 ~]$ sudo mysql -h database-1.cebmq8qksdt.us-east-1.rds.amazonaws.com -u admin -P 3306 -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MySQL connection id is 21  
Server version: 8.0.42 Source distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MySQL [(none)]> [REDACTED]
```

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

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

	<p>A. Create a database called webappdb with the following command using the MySQL CLI:</p> <pre><code>CREATE DATABASE webappdb;</code></pre> <p>You can verify that it was created correctly with the following command:</p> <pre><code>SHOW DATABASES;</code></pre> <ol style="list-style-type: none"> 1. Create a data table by first navigating to the database we just created: <pre><code>USE webappdb;</code></pre> <p>Then, create the following transactions table by executing this create table command:</p> <pre><code>CREATE TABLE IF NOT EXISTS transactions (id INT NOT NULL AUTO_INCREMENT, amount DECIMAL(10,2), description VARCHAR(100), PRIMARY KEY (id));</code></pre> <p>Verify the table was created:</p> <pre><code>SHOW TABLES;</code></pre> <ol style="list-style-type: none"> 2. Insert data into the table for use/testing later: <pre><code>INSERT INTO transactions (amount, description) VALUES ('400', 'groceries');</code></pre> <p>Verify that your data was added by executing the following command:</p> <pre><code>SELECT * FROM transactions;</code></pre> <ol style="list-style-type: none"> 3. When finished, just type exit and hit enter to exit the MySQL client.
--	---

Create S3 Bucket & Create IAM Role

- A . Create S3 bucket with default setting
- B. Create IAM Role for EC2 ----> the policies needs to be attached
 - **AmazonSSMManagedInstanceCore**,
 - **AmazonS3ReadOnlyAccess**

 Step-by-Step: Create IAM Role for EC2 (S3 Read-Only Access)

Step 1 – Navigate to IAM

1. Log in to the AWS Console and search for IAM.
2. In the left-hand navigation pane, click Roles.

IAM Dashboard info

IAM resources

Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
1	1	6	1	0

What's new View all

Updates for features in IAM

- Amazon Bedrock introduces API keys for streamlined development. 1 month ago
- AWS Service Reference Information now supports annotations for service actions. 1 month ago
- AWS expands resource control policies (RCPs) support to two additional services. 1 month ago
- AWS IAM now enforces MFA for root users across all account types. 2 months ago

▼ more

AWS Account

Account ID
941098798453

Account Alias
Create

Sign-In URL for IAM users in this account
<https://941098798453.signin.aws.amazon.com/console>

Quick Links

[My security credentials](#)

Manage your access keys, multi-factor authentication (MFA) and other credentials.

Step 2 – Create New Role

1. Click Create role.
2. Under Trusted entity type, select AWS service.
3. Choose EC2 as the use case (this allows the role to be assumed by EC2 instances).
4. Click Next.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2

Choose a use case for the specified service

Use case

EC2
Allows EC2 instances to call AWS services on your behalf.

EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

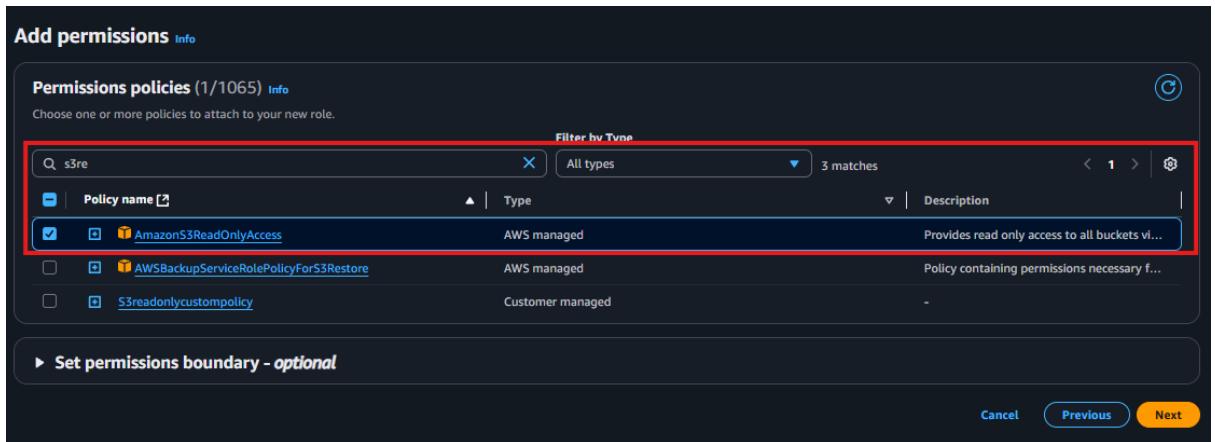
EC2 - Spot Instances
Allows EC2 spot Instances to launch and manage spot instances on your behalf.

EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.

Step 3 – Attach Permission Policy

1. In the policy list, search for AmazonS3ReadOnlyAccess.
 2. Select the checkbox for AmazonS3ReadOnlyAccess.
 3. Click Next.
- This policy allows the EC2 instance to list and retrieve objects from any S3 bucket.



Step 4 – Name and Tag the Role

1. Provide a name, e.g. EC2-S3ReadOnlyRole.
2. (Optional) Add tags such as:
 - o Key = Project, Value = 3TierArchitecture
3. Click Create role.

The role is now created.

Step 5 – Attach the Role to EC2 Instances

1. Navigate to EC2 > Instances.
 2. Select the Web or App instance that requires S3 access.
 3. Click Actions → Security → Modify IAM role.
 4. In the dropdown, select EC2-S3ReadOnlyRole.
 5. Click Update IAM role.
- Upload your aws-three-tier-web-architecture-workshop → application-code then upload the app-tier, web-tier, nginx.conf.

Configure App-Server

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

And add the host name username & Password--> Then Save it

```
module.exports = Object.freeze({
  DB_HOST : 'database-1.cxz1od1wh1ta.us-east-1.rds.amazonaws.com',
  DB_USER : 'admin',
```

```

    DB_PWD : 'Admin123',
    DB_DATABASE : 'webappdb'
});

aws s3 cp s3://mynewbsocjod/app-tier/ . --recursive
s3://mynewbsocjod/web-tier/

```

- After Updating the credentials Updating the credentials to your Dbconfig.json upload the App_tier folder to your S3 Bucket
 1. Open S3 buckeg what you created in step 1
 2. Choose upload
 3. Select the App-tier folder only
 4. Then Upload the folder
- Go Back to your App-Tier Putty Session
 - a. Start by installing NVM (node version manager).

```

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh |
bash
source ~/.bashrc

```

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

```

nvm install 16
nvm use 16

```

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

```

npm install -g pm2

```

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

```

cd ~/
aws s3 cp s3://BUCKET_NAME/app-tier/ app-tier --recursive

```

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

```

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

```

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

```

pm2 list

```

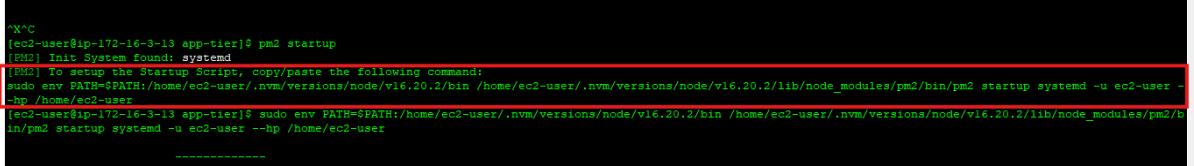
G: you see a status of online, the app is running. If you see errored, then you need to do some troubleshooting. To look at the latest errors, use this command:

pm2 logs

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

pm2 startup

After that run the command from the terminal



```
^X^C
[ec2-user@ip-172-16-3-13 app-tier]$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user --hp /home/ec2-user
[ec2-user@ip-172-16-3-13 app-tier]$ sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user --hp /home/ec2-user
-----
```

After you run it, save the current list of node processes with the following command:

```
pm2 save
```

Test App Tier

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

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

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

The response should look like the following:

"This is the health check"

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

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

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

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

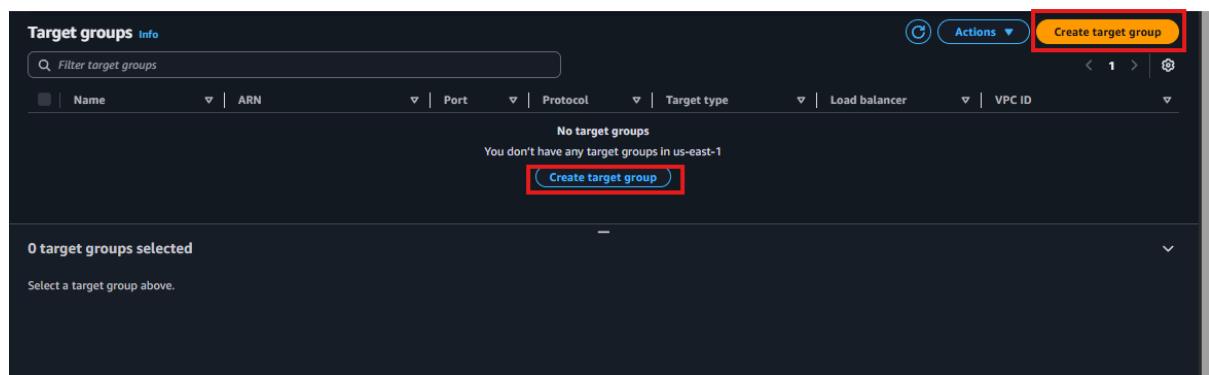
If you see both of these responses, then your networking, security, database and app configurations are correct.

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

Step 10 – Internal load balancer and Auto scaling the app-server

Step 1 – Create Target Group (App Tier)

1. Go to EC2 → Target Groups in the AWS Console.
2. Click Create target group.



3. Configure the following options:

Field	Value
Target type	Instance
Name	app-tier-target-group
Protocol	HTTP
Port	4000
VPC	Select your custom VPC
Health Check Protocol	HTTP
Health Check Path	/health

4. Click Next.
5. Do not register any targets yet (leave it empty).

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Protocol for load balancer-to-target communication. Can't be modified after creation.

Port

Port number where targets receive traffic. Can be overridden for individual targets during registration.

IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-07a4339e7f23981e0 (3-tier-vpc)
172.16.0.0/16

[Create VPC](#)

Protocol version

HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/health

Up to 1024 characters allowed.

▼ Advanced health check settings

[Restore defaults](#)

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

- Traffic port**
- Override**

Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

2

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

2 seconds

2-120

Interval

The approximate amount of time between health checks of an individual target.

5 seconds

5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

200

Available instances (2)

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Instance ID	Name	State	Security groups	Zone	Private IPv4
i-04110f62f37f4b6f	app-server	Running	app-server	us-east-1a	172.16.3.13
i-0c351a6b00ea73d1b	jump-server	Running	Jump-server	us-east-1a	172.16.1.16

0 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

4000
1-65535 (separate multiple ports with commas)

Include as pending below

Review targets

Targets (0)

No instances added yet
Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending

Create target group

- Click Create target group.

Step 2 – Create Internal Application Load Balancer

- Go to EC2 → Load Balancers.

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Create load balancer

No load balancers
You don't have any load balancers in us-east-1

Create load balancer

0 load balancers selected

Select a load balancer above.

- Click Create Load Balancer → Application Load Balancer.



3. Configure the following:

Field	Value
Name	internal-app-alb
Scheme	Internal
IP address type	IPv4
VPC	Select your custom VPC
Availability Zones	Select Private subnets (App subnets)

Click Next: Configure Security Settings (leave default).

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type [Info](#)
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4

- Includes only IPv4 addresses.

Dualstack

- Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

[Create VPC](#)

IP pools - new [Info](#)
You can optionally choose to configure an IPAM pool as the preferred source for your load balancer's IP addresses. Create or view Pools in the [Amazon VPC IP Address Manager console](#).

Use IPAM pool for public IPv4 addresses
Compatible with Internet-facing scheme, IPv4 and Dualstack IP address types.

Availability Zones and subnets [Info](#)
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az4)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
 [private-subnet-01](#)

us-east-1b (use1-az6)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
 [private-subnet-02](#)

Security groups [Info](#)
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
 [Remove](#) [Create security group](#)

Listeners and routing [Info](#)
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.

Listener HTTP:80

Protocol	Port
HTTP	80
1-65535	

Default action [Info](#)

Forward to	HTTP
app-server-target-group	HTTP
Target type: Instance, IPv4	

[Create target group](#) [Remove](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)
You can add up to 49 more listeners.

Click **Next** and **Create Load Balancer**.

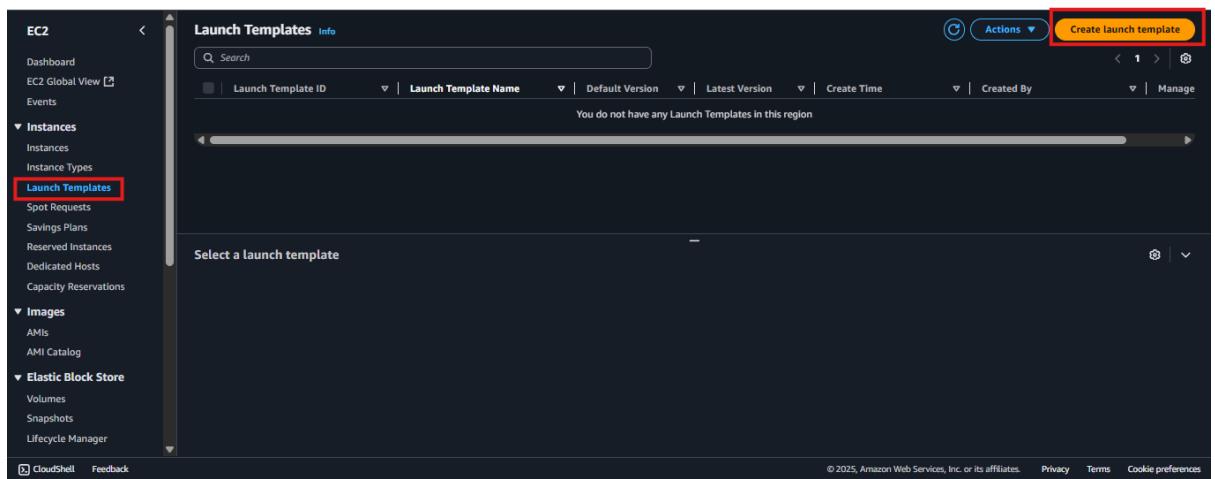
Step 1 – Create an AMI from the App Server

1. Navigate to EC2 > Instances.
2. Select your existing App EC2 instance (the one you've already configured and tested).

3. Click Actions → Image → Create Image.
4. Provide a name (e.g. *app-server-ami*) and description.
5. Leave other values as default and click Create Image.
6. Wait a few minutes until the AMI status becomes Available (check under AMI section in the EC2 console).

Step 2 – Create a Launch Template Using the AMI

1. Go to EC2 → Launch Templates → Create launch template



2. Set the following:

Field	Value
Launch template name	app-tier-template
AMI ID	Select the AMI created in Step 1
Instance type	(same type used for the App Server, e.g. t2.micro)
Key pair	Select the required key pair
Security group	App Server SG (allows port 4000 from the internal ALB)

Field	Value

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

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '=', '@'.

Template version description

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

▼ Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents **My AMIs** **Quick Start**

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

Specify a custom value...

app-server-image ami-0cccd1e0814c607efd
2025-08-17T15:27:35.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

app-server-image ami-0cccd1e0814c607efd
2025-08-17T15:27:35.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description
app-server-image

Architecture **AMI ID**
x86_64 ami-0cccd1e0814c607efd

Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro **Free tier eligible**

Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

backup-server

[Create new key pair](#)

Network settings [Info](#)

Subnet | [Info](#)

Don't include in launch template

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

[Create new subnet](#)

Availability Zone | [Info](#)

Don't include in launch template

[Enable additional zones](#)

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

app-server sg-034f16022939e2a29 X

VPC: vpc-0784539e7f23981e0

[Compare security group rules](#)

[Advanced network configuration](#)

Storage (volumes) [Info](#)

EBS Volumes

Volume 1 (AMI Root) : 8 GiB, EBS, General purpose SSD (gp3), 3000 IOPS
AMI Volumes are not included in the template unless modified

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

[Add new volume](#)

Advanced details [Info](#)

IAM instance profile [Info](#)

S3readonlyazaccess
am:aws:iam::941098798453:instance-profile/S3readonlyazaccess

[Create new IAM profile](#)

Hostname type [Info](#)

Don't include in launch template

DNS Hostname [Info](#)

Enable resource-based IPv4 (A record) DNS requests

Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)

Don't include in launch template

Shutdown behavior [Info](#)

Don't include in launch template

Stop - Hibernate behavior [Info](#)

Don't include in launch template

Termination protection [Info](#)

- Click Create launch template

🔧 Step-by-Step: App Target Group + Internal Load Balancer (Port 4000)

Step 3 – Create Auto Scaling Group Based on the Launch Template

- Navigate to Auto Scaling Groups → Create Auto Scaling group

Amazon EC2 Auto Scaling
helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

[Create Auto Scaling group](#)

How it works

Auto Scaling group

Minimum size

Scale out or needed

Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started [\[?\]](#)

[What is Amazon EC2 Auto Scaling?](#)

- Auto Scaling group name: app-tier-asg
- Launch template: select app-tier-template

Name

Auto Scaling group name
Enter a name to identify the group.
 Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

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

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
 (C)

[Create a launch template](#) (A)

Version

(C)

[Create a launch template version](#) (A)

4. VPC: Select the custom VPC

5. Subnets: Select private subnets

Network Info

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

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
 (C)

[Create a VPC](#) (A)

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

(C)

(C)

(C)

[Create a subnet](#) (A)

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Click Next

Step 4 – Attach to the Internal Load Balancer

1. Select **Attach to an existing load balancer**
2. Choose **Application Load Balancer**
3. Select **internal-app-alb**

- Select the target group you created for the App Tier (e.g. app-target-group)

Integrate with other services - optional Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info

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

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

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

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

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 ▼ 

app-server-target-group | HTTP 

Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▾

Desired capacity
Specify your group size.
2

Scaling Info

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

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

Min desired capacity 2 **Max desired capacity** 3

Equal or less than desired capacity Equal or greater than desired capacity

Automatic scaling - optional

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

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

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

Scaling policy name
Target Tracking Policy

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

Average CPU utilization ▾

Target value
80

Click **Next** and **finish** the creation of auto-scaling group

Step 11 – Launch EC2 Instances (Web-Tier)

- Repeat the steps for launching new EC2 instances, but select the public subnets. And choose the web-tier security group that we have created early. And enable public IP assigning

▼ Network settings [Info](#)

VPC - required | [Info](#)

vpc-07a4339e7f23981e0 (3-tier-vpc)
172.16.0.0/16

Subnet | [Info](#)

subnet-021b4ea9190085709 public-subnet-01
VPC: vpc-07a4339e7f23981e0 Owner: 941098798453 Availability Zone: us-east-1a (use1-az4)
Zone type: Availability Zone IP addresses available: 249 CIDR: 172.16.1.0/24

Create new subnet

Auto-assign public IP | [Info](#)

Enable

Additional charges apply when outside of free tier allowance

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

web-server sg-010e530c8c367d0d5 X
VPC: vpc-07a4339e7f23981e0

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

▼ Advanced details [Info](#)

Domain join directory | [Info](#)

Select [Create new directory](#)

IAM instance profile | [Info](#)

S3readonlyazaccess
arn:aws:iam::941098798453:instance-profile/S3readonlyazaccess

Hostname type | [Info](#)

IP name

DNS Hostname | [Info](#)

Enable IP name IPv4 (A record) DNS requests
 Enable resource-based IPv4 (A record) DNS requests
 Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery | [Info](#)

Select

Shutdown behavior | [Info](#)

Stop

- Then click on “**Launch Instance**”

Update Config File

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

```

40     listen      [::]:80;
41     server_name _;
42
43     #health check
44     location /health {
45         default_type text/html;
46         return 200 "<!DOCTYPE html><p>Web Tier Health Check</p>\n";
47     }
48
49     #react app and front end files
50     location / {
51         root    /home/ec2-user/web-tier/build;
52         index  index.html index.htm;
53         try_files $uri /index.html;
54     }
55
56     #proxy for internal lb
57     location /api/{
58         proxy_pass http://[REPLACE-WITH-INTERNAL-LB-DNS]:80; ←
59     }
60
61 }
62
63
64 # Settings for a TLS enabled server.
65 #
66 #   server {
67 #       listen      443 ssl http2;
68 #       listen      [::]:443 ssl http2;
69 #       server_name _;
70 #       root        /usr/share/nginx/html;
71 #

```

<http://internal-pvt-lb-1627883891.ap-south-1.elb.amazonaws.com/>:

From <<http://internal-pvt-lb-1627883891.ap-south-1.elb.amazonaws.com/>>

Configure Web Instance

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

```

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

```

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

```

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

```

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

```

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

```

1. 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.

```
1 sudo yum install nginx1 -y
```

1. We will now have to configure NGINX. Navigate to the Nginx configuration file with the following commands and list the files in the directory:

```

1 cd /etc/nginx
2 ls

```

You should see an nginx.conf file. We're going to delete this file and use the one we uploaded to s3. 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://BUCKET_NAME/nginx.conf .
```

Then, restart Nginx with the following command:

```
1 sudo service nginx restart
```

To make sure Nginx has permission to access our files execute this command:

```
1 chmod -R 755 /home/ec2-user
```

And then to make sure the service starts on boot, run this command:

```
1 sudo chkconfig nginx on
```

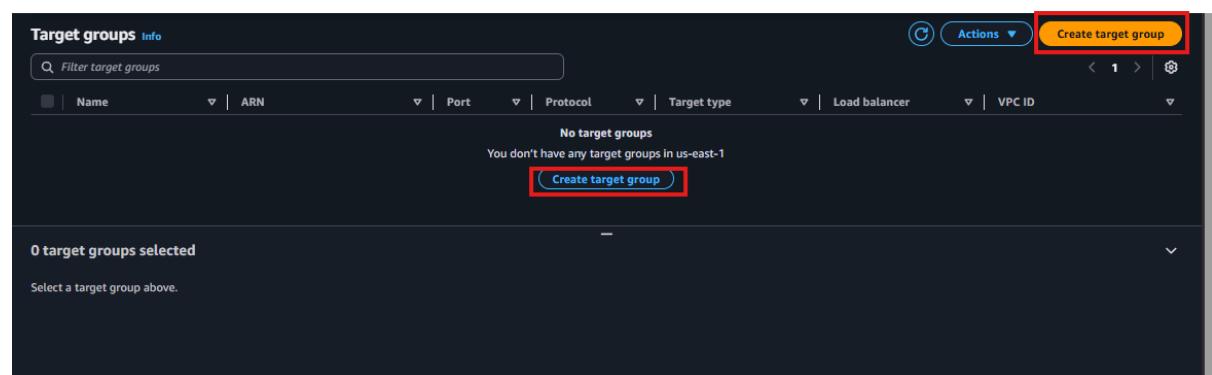
Now when you plug in the public IP of your web tier instance, you should see your website, which you can find on the Instance details page on the EC2 dashboard. If you have the database connected and working correctly, then you will also see the database working. You'll be able to add data. Careful with the delete button, that will clear all the entries in your database.

Step 12 – External load balancer and Auto scaling the web-server

Step 1 – Create Target Group (Web-Tier)

4. Go to EC2 → Target Groups in the AWS Console.

5. Click Create target group.



6. Configure the following options:

Field	Value
Target type	Instance
Name	web-tier-target-group

Field	Value
Protocol	HTTP
Port	80
VPC	Select your custom VPC
Health Check Protocol	HTTP
Health Check Path	/health

7. Click Next.

8. Do not register any targets yet (leave it empty).

Target group name
web-tier-target-group

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.
HTTP

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.
80

IP address type
Only targets with the indicated IP address type can be registered to this target group.
 IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
 IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.
vpc-07a4339e7f23981e0 (3-tier-vpc)
172.16.0.0/16

Protocol version
 HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
 HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/health

Up to 1024 characters allowed.

Advanced health check settings

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

Traffic port

Override

Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

2

2-10

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

2 seconds

2-120

Interval

The approximate amount of time between health checks of an individual target.

5 seconds

5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

200

Available instances (5)

Instance ID	Name	State	Security groups	Zone	Private IPv4
I-074de6b62e7ec0d30	app-server	Running	app-server	us-east-1a	172.16.3.17
I-09c23b62a3070f7cc	app-server	Running	app-server	us-east-1b	172.16.4.12
I-0f21ca49282298cd	web-server	Running	web-server	us-east-1a	172.16.1.15
I-0411f0f62f37f4bf	app-server	Running	app-server	us-east-1a	172.16.3.13
I-0c351a6b00ea73d1b	jump-server	Running	Jump-server	us-east-1a	172.16.1.16

0 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Review targets

Targets (0)

Filter targets

Show only pending

No instances added yet

Specify instances above, or leave the group empty if you prefer to add targets later.

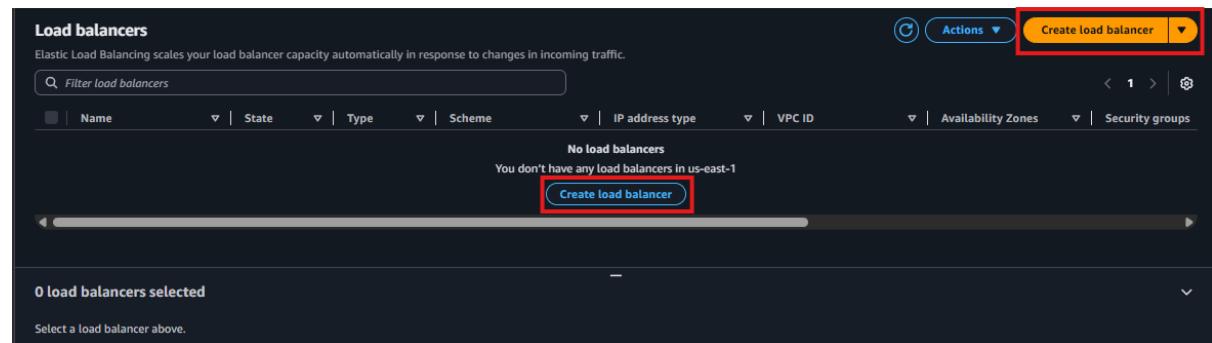
0 pending

Cancel Previous Create target group

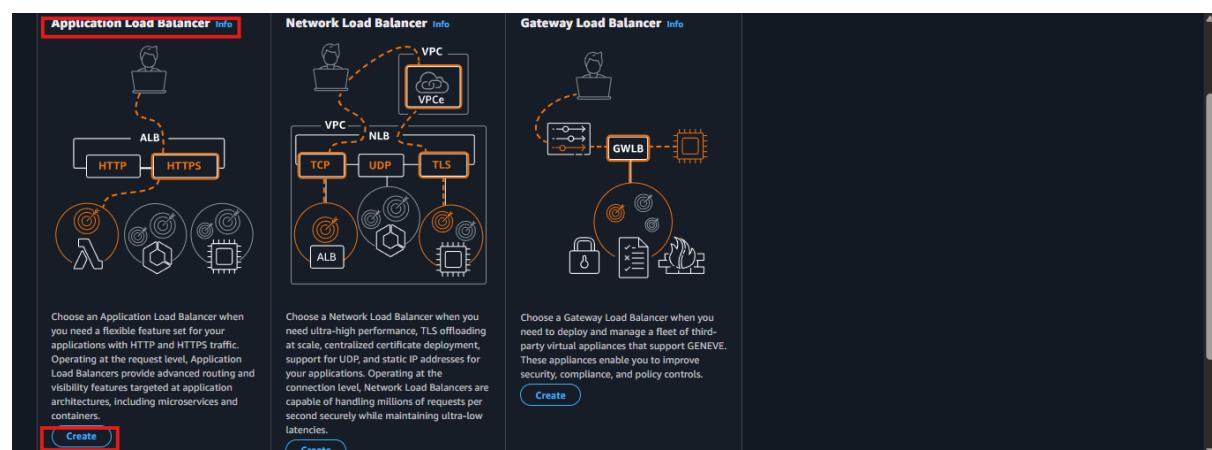
9. Click Create target group.

Step 2 – Create Internal Application Load Balancer

4. Go to EC2 → Load Balancers.



5. Click Create Load Balancer → Application Load Balancer.



6. Configure the following:

Field	Value
Name	external-app-alb
Scheme	Internet-facing
IP address type	IPv4

Field	Value
VPC	Select your custom VPC
Availability Zones	Select Public subnets (Web subnets)

Click Next: Configure Security Settings (leave default).

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

Scheme Info
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only.

Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups ?

▼

IP pools - new Info
You can optionally choose to configure an IPAM pool as the preferred source for your load balancer's IP addresses. Create or view Pools in the Amazon VPC IP Address Manager console ?

Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets Info
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az4)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
 ▼

us-east-1b (use1-az6)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
 ▼

The screenshot shows the AWS Load Balancer configuration interface. In the 'Security groups' section, a security group named 'external-load-balancer' is selected. Under 'Listeners and routing', a new listener for port 80 is being configured. The 'Protocol' is set to 'HTTP' and the 'Port' is '80'. The 'Default action' is set to 'Forward to' a target group named 'web-tier-target-group'. This target group is highlighted with a red box. A dropdown menu for 'In use' shows another target group named 'app-server-target-group'. At the bottom, there is an 'Add listener tag' button.

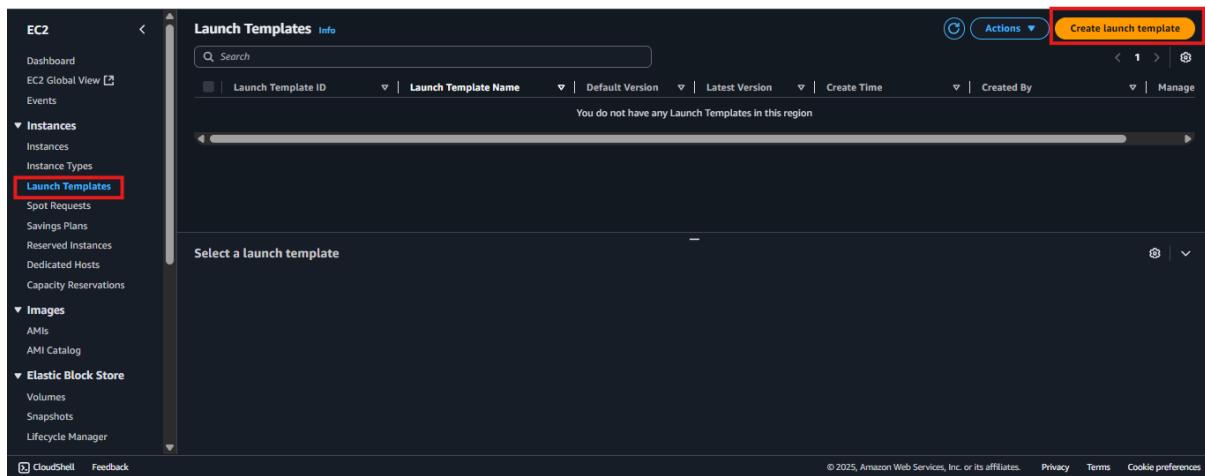
Click **Next** and **Create Load Balancer**.

🔧 Step 1 – Create an AMI from the App Server

1. Navigate to EC2 > Instances.
2. Select your existing Web EC2 instance (the one you've already configured and tested).
3. Click Actions → Image → Create Image.
4. Provide a name (e.g. *web-tier-ami*) and description.
5. Leave other values as default and click Create Image.
6. Wait a few minutes until the AMI status becomes Available (check under AMI section in the EC2 console).

🔧 Step 2 – Create a Launch Template Using the AMI

4. Go to EC2 → Launch Templates → Create launch template



5. Set the following:

Field	Value
Launch template name	web-tier-template
AMI ID	Select the AMI created in Step 1
Instance type	(same type used for the web Server, e.g. t2.micro)
Key pair	Select the required key pair
Security group	web Server SG (allows port 80 from the External ALB)

Launch template name and description

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

Template version description
 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](#)

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[Recents](#) [My AMIs](#) [Quick Start](#)

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

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

web-tier-template	ami-0ecceab054bcfc469	2025-08-17T18:43:16.000Z	Virtualization: hvm	ENAv enabled: true	Root device type: ebs	Boot mode: uefi-preferred
-------------------	-----------------------	--------------------------	---------------------	--------------------	-----------------------	---------------------------

Description
web-server-ami

Architecture x86_64 AMI ID ami-0ecceab054bcfc469

Instance type [Info](#) | [Get advice](#)

Advanced

Instance type

t2.micro **Free tier eligible**

Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

backup-server

[Create new key pair](#)

Network settings [Info](#)

Subnet | [Info](#)

Don't include in launch template

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

[Create new subnet](#)

Availability Zone | [Info](#)

Don't include in launch template

[Enable additional zones](#)

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

external-load-balancer sg-0255a3f2a977ddb94 X

VPC: vpc-07a4339e/f23981e0

[Compare security group rules](#)

[Advanced network configuration](#)

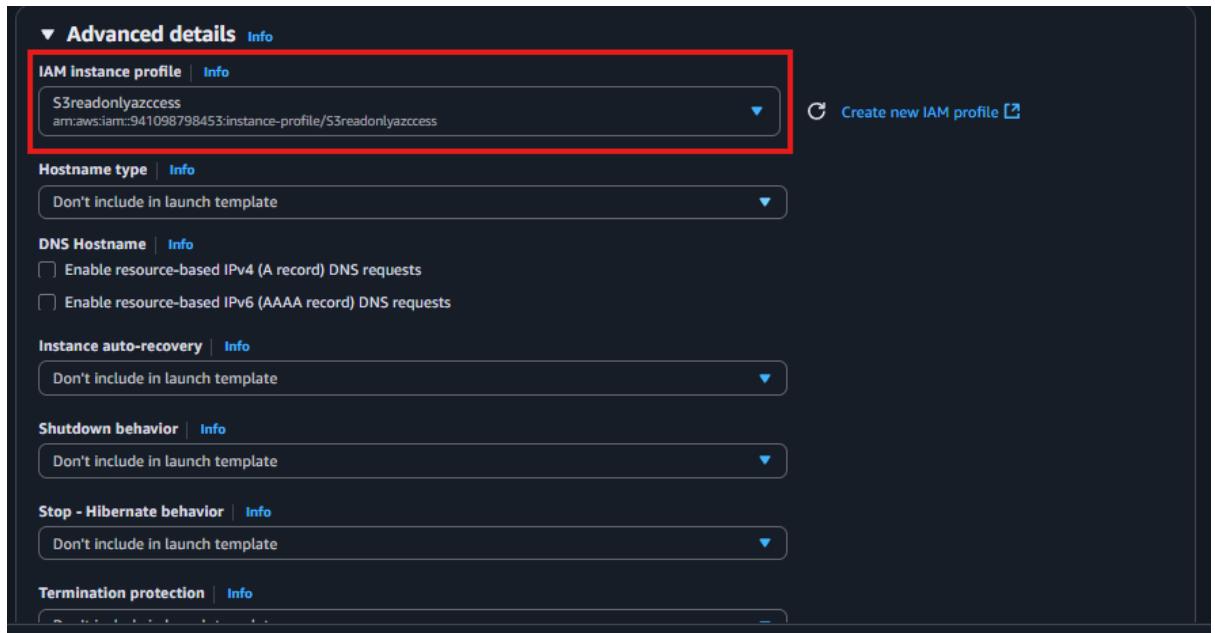
Storage (volumes) [Info](#)

EBS Volumes

Volume 1 (AMI Root) : 8 GiB, EBS, General purpose SSD (gp3), 3000 IOPS
AMI Volumes are not included in the template unless modified

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

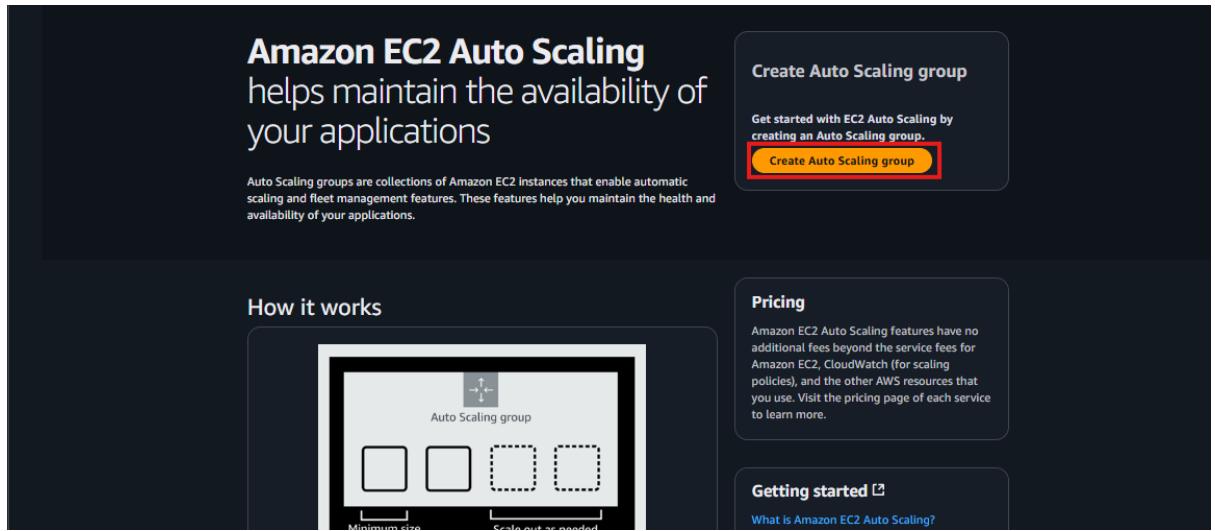
[Add new volume](#)



6. Click Create launch template

Step 3 – Create Auto Scaling Group Based on the Launch Template

6. Navigate to **Auto Scaling Groups** → **Create Auto Scaling group**



7. **Auto Scaling group name:** web-tier-asg

8. **Launch template:** select web-tier-template

Name

Auto Scaling group name
Enter a name to identify the group.
web-tier-asg
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

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

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
web-tier-template (C)

Create a launch template [?]

Version
Default (1) (C)

Description
web-tier-template

Launch template web-tier-template lt-089291fa8d517e2da	Instance type t2.micro
AMI ID ami-0ecceab054bcfc469	Security groups -
	Request Spot Instances No

9. VPC: Select the custom VPC

10. Subnets: Select Public subnets

Network Info

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

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-07a4339e7f23981e0 (3-tier-vpc) (C)
172.16.0.0/16

Create a VPC [?]

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

Select Availability Zones and subnets (C)

use1-az4 (us-east-1a) | subnet-021b4ea9190085709 (public-subnet-01) X
172.16.1.0/24

use1-az6 (us-east-1b) | subnet-0a1bc6fe3215ace94 (public-subnet-02) X
172.16.2.0/24

Create a subnet [?]

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Click Next

Step 4 – Attach to the External Load Balancer

5. Select **Attach to an existing load balancer**

6. Choose **Application Load Balancer**

7. Select **external-app-alb**

8. Select the **target group** you created for the Web Tier (e.g. Web-target-group)

Integrate with other services - optional Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info

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

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

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

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

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 ▼ 

web-tier-target-group | HTTP 
Application Load Balancer: external-app-alb

Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▼

Desired capacity

Specify your group size.

2

Scaling Info

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

Scaling limits

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

Min desired capacity 2 Equal or less than desired capacity

Max desired capacity 3 Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy Info

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

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

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

Scaling policy name
Target Tracking Policy

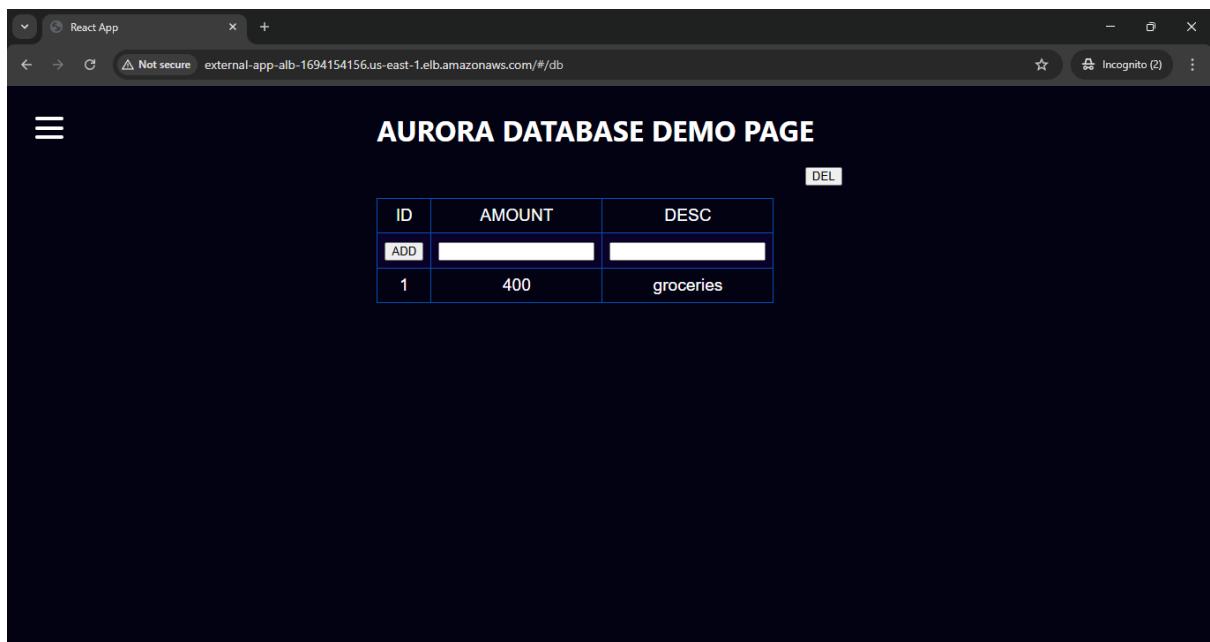
Metric type Info

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

Average CPU utilization ▼

Target value
80

Click **Next**, then **finish** the creation. Once the auto-scaling group created copy the external load balancer Domain name and paste it into browser and press enter. Output must be the web page that is just configured. The output look like this



The request may still be in HTTP. But the application is working fine.

Step - 13 Configure CloudFront → External Load Balancer (with SSL)

Step 1 – Prepare Your External Load Balancer Domain

Make sure you have the DNS name of your **External Application Load Balancer** (e.g. external-web-alb-1234567890.us-east-1.elb.amazonaws.com).

Step 2 – Create an SSL Certificate in ACM

1. Navigate to **AWS Certificate Manager (ACM)** in N.Virginia where your **CloudFront distribution will run** (typically us-east-1).

AWS Certificate Manager (ACM) Certificates (2/2)

Certificate ID	Domain name	Type	Status	In use	Renewal eligibility
fade89ff-95c-47d6-96be-e23793085c89	realdynamic.myawsdomain.shop	Amazon Issued	Issued	No	Ineligible
b6bab9c9-7072-4011-a100-e12859f849ba	*.myawsdomain.shop	Amazon Issued	Issued	No	Ineligible

2. Click Request a certificate → Request a public certificate

Request certificate

Certificate type Info
ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit [AWS Private Certificate Authority](#).

Cancel Next

3. Enter your domain name (e.g. www.example.com or app.example.com)

Domain names

Provide one or more domain names for your certificate.

Fully qualified domain name Info

Add another name to this certificate
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Allow export Info
 Disable export
Use this certificate only with integrated AWS services. The private key for this certificate will be disallowed for exporting from AWS.
 Enable export
Export this certificate and private key for use with any TLS workflow. ACM will charge your account based on the requested domains when the certificate is issued for the first time and for each renewal.

Validation method Info
Select a method for validating domain ownership.

DNS validation - recommended
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

Email validation
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

4. Complete the validation by adding the provided CNAME record in Route 53

Domains (1)

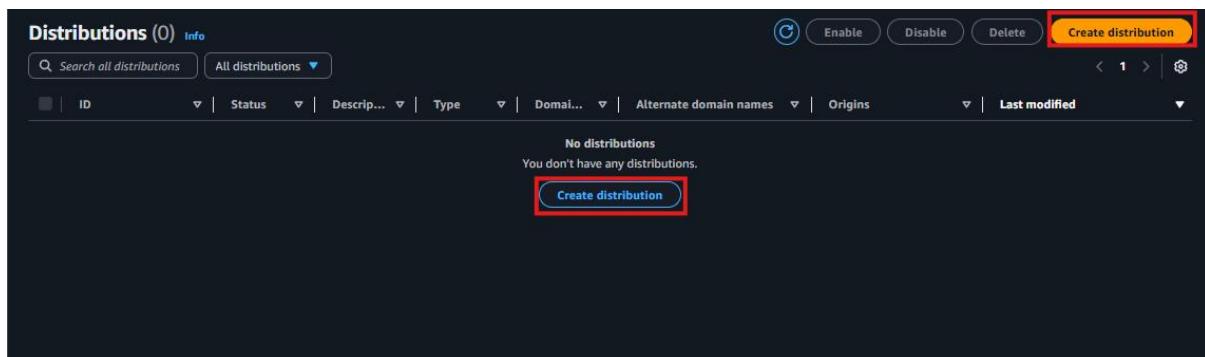
Domain	Status	Renewal status	Type	CNAME name	CNAME value
appreact.myawsdomain.shop	Pending validation	-	CNAME	_d7c4ad481dcad65edd677ec8096e39a0.appreact.myawsdomain.shop.	_29e069validation



5. After validation, the certificate state becomes **Issued**

Step 3 – Create CloudFront Distribution

1. Open the **CloudFront console** and click **Create Distribution**



Get started

Connect your websites, apps, files, video streams, and other content to CloudFront. We optimize the performance, reliability, and security for your web traffic.

Distribution options Info

Distribution name

Name will be stored as a tag on the resource. You can add a name, or more tags, later.

web-server

Description - optional

Distribution type

Single website or app

Choose if each website or application will have a unique configuration.

Multi-tenant architecture - New

Choose when you have multiple domains that need to share configurations. This is a common architecture for SaaS providers.

Custom domain Info

Domain - optional

Use your own custom domain with free HTTPS to provide a secure, friendly URL for your app. You can add a custom domain later if you do not have a Route 53 zone in this account.

2. Under **Origin configuration** specify:

Origin type

Your origin is where your content (such as a website or app) lives. CloudFront works with AWS-based origins and origins hosted on other cloud providers.

<input type="radio"/> Amazon S3 Deliver static assets like files and images, statically generated websites or single page applications (SPA).	<input checked="" type="radio"/> Elastic Load Balancer Deliver applications hosted behind ELB such as dynamic websites, web services, and APIs.	<input type="radio"/> API Gateway Deliver API endpoints for REST APIs hosted on API Gateway.
<input type="radio"/> Elemental MediaPackage Deliver end-to-end live events or video on demand (VOD).	<input type="radio"/> VPC origin Deliver applications and content hosted within private VPCs, such as EC2 instances and Application Load Balancers.	<input type="radio"/> Other Refer to any AWS or non-AWS origin through its publicly resolvable URL.

Origin

Elastic Load Balancing origin
Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

external-app-alb-1694154156.us-east-1.elb.amazonaws.com

[Browse load balancers](#)

Origin path - optional
The directory path within your origin where your content is stored. [Learn more](#)

/path

Enable security

Web Application Firewall (WAF) [Info](#)

<input type="radio"/> Enable security protections Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.	<input checked="" type="radio"/> Do not enable security protections Select this option if your application does not need security protections from AWS WAF.
---	--

[Cancel](#) [Previous](#) [Next](#)

- Create a **Distribution**. Then go to “**Origins**” and click “**Edit**”.

web-server Standard [View metrics](#)

[General](#) [Security](#) [Origins](#) [Behaviors](#) [Error pages](#) [Invalidations](#) [Tags](#) [Logging](#)

Origins (1/1)

Origin name	Origin domain	Origin path	Origin type	Origin Shield
external-app-alb-1694154156.us-east-1.elb.amazonaws.com-meg66rwg4zu	external-app-alb-1694...		Elastic Load Balancing	-

[Edit](#) [Delete](#) [Create origin](#)

Origin groups (0)

Origin group name	Origins	Failover criteria
No origin groups You don't have any origin groups.		

[Create origin group](#)

- Select port for the Origin domain in this external load balancer’s .

Settings

Origin domain
Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.

Protocol [Info](#)

HTTP only

HTTPS only

Match viewer

HTTP port
Enter your origin's HTTP port. The default is port 80.

Origin path - optional
Enter a URL path to append to the origin domain name for origin requests.

Name
Enter a name for this origin.

- Finally click on “Save Changes”.

Field	Value
Origin domain	<external-web-alb-dns>
Origin protocol	HTTPS only (recommended)
Origin path	<i>leave blank</i>
Origin ID	external-alb-origin

- Once distribution created click on add domain enter your SSL certificate name just before created. Then click next and select your certificate finally press add domain.

web-server Standard [View metrics](#)

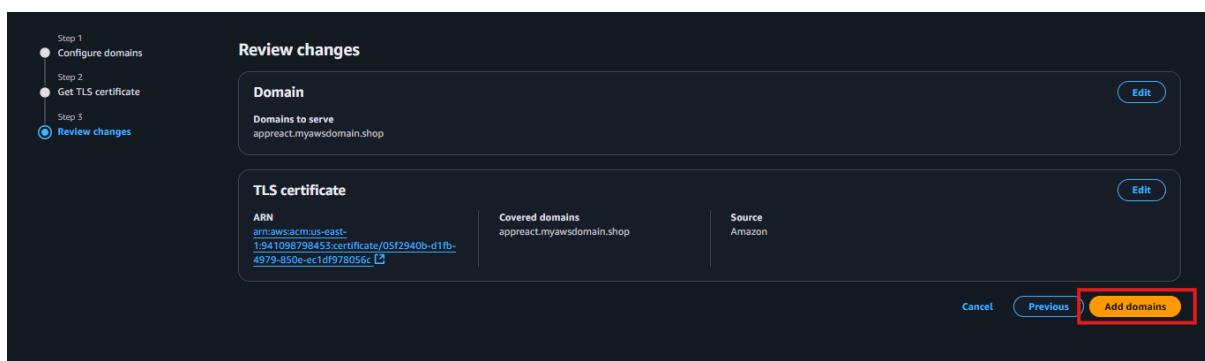
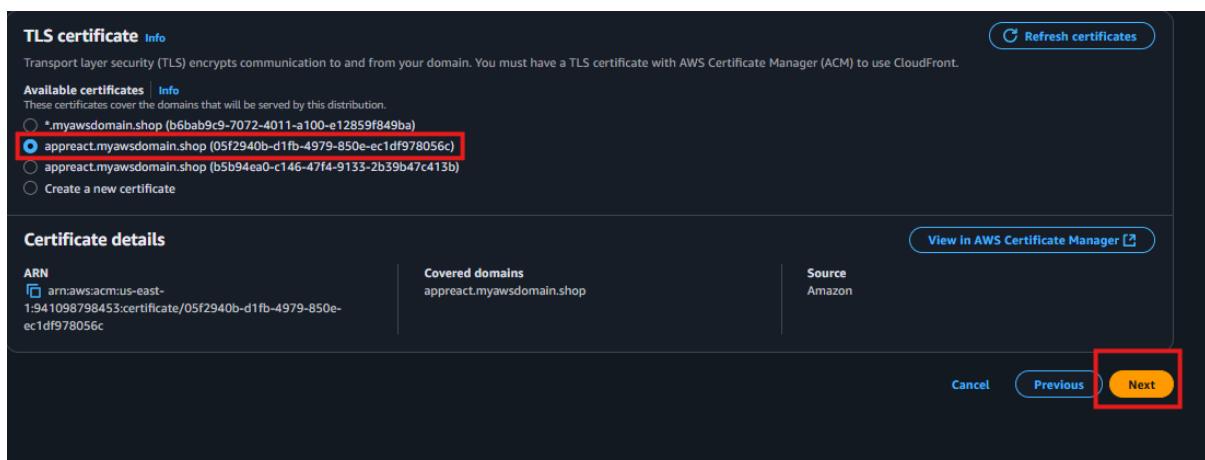
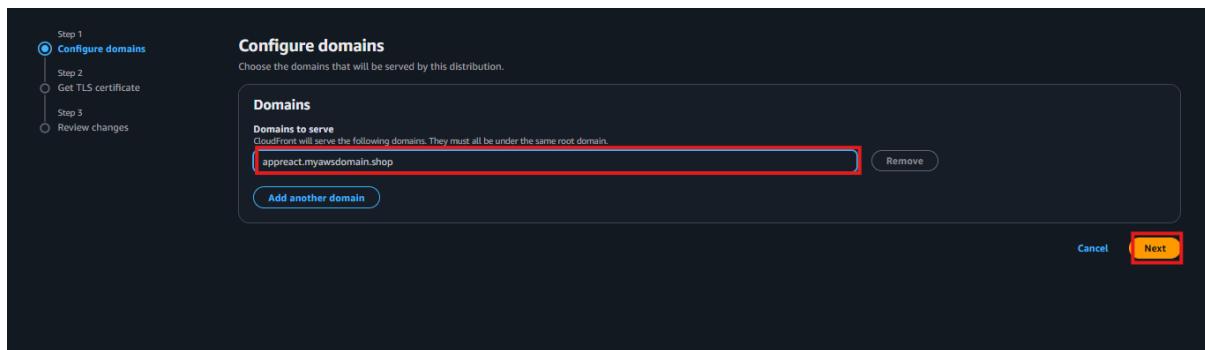
[General](#) [Security](#) [Origins](#) [Behaviors](#) [Error pages](#) [Invalidations](#) [Tags](#) [Logging](#)

Details

Name: web-server	Distribution domain name: d1f2hers3ou4cm.cloudfront.net	ARN: arn:aws:cloudfront::941098798453:distribution/EJ2PMEW5TB1Q	Last modified: Deploying
------------------	---	---	--------------------------

Settings

Description: -	Alternate domain names: Add domain	Standard logging: Off
Price class: Use all edge locations (best performance)	Cookie logging: Off	Default root object: -
Supported HTTP versions: HTTP/2, HTTP/1.1, HTTP/1.0		



Step – 14 Final Step – Create DNS Record in Route 53

- Open the Route 53 console → Hosted Zones

The screenshot shows the Route 53 Dashboard. On the left, there's a navigation sidebar with options like 'Route 53', 'Dashboard', 'Hosted zones' (which is selected and highlighted with a red box), 'IP-based routing', 'Traffic flow', 'Domains', 'Resolver', and 'CloudShell'. The main content area has four sections: 'DNS management' (with a red box around it), 'Traffic management', 'Availability monitoring', and 'Domain registration'. Below these are sections for 'Register domain' and 'Notifications'.

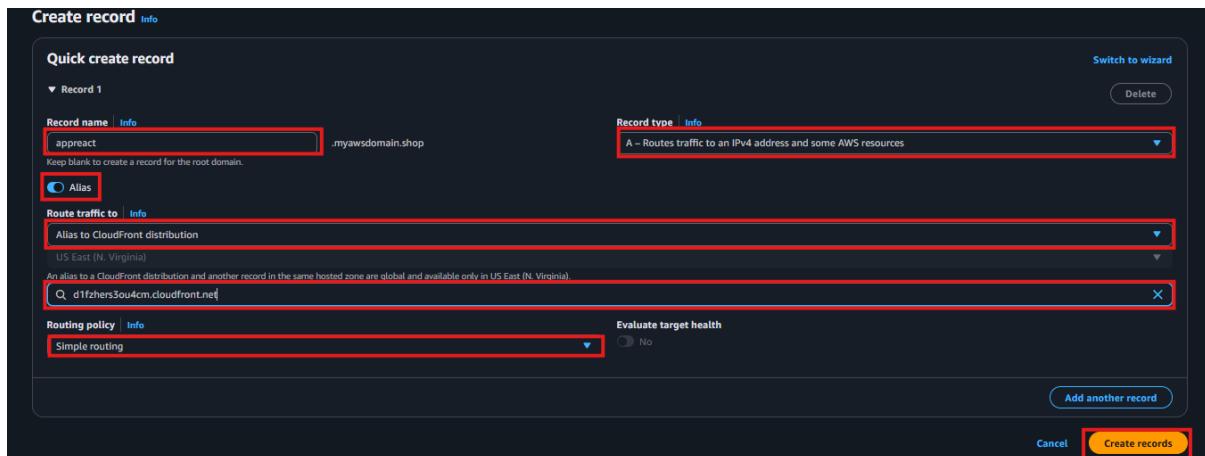
② Select the hosted zone for your domain (e.g. example.com)

The screenshot shows the 'Hosted zones' page for the domain 'myawsdomain.shop'. It lists one hosted zone: 'myawsdomain.shop' (Public, Created by Route 53, Record count 6). The 'Hosted zone name' column is highlighted with a red box around the entry 'myawsdomain.shop'.

③ Click Create record

The screenshot shows the 'Hosted zone details' page for 'myawsdomain.shop'. It has tabs for 'Records (6)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (6)' tab is active. A red box highlights the 'Create record' button at the top right of the records table. The table lists six records: NS, SOA, CNAME, CNAME, CNAME, and CNAME. The 'Create record' button is also highlighted with a red box.

④ Enter the domain name (SSL certificate name) and record type, click the Alias button and enter the details finally click “Create Records”.



We have successfully deployed a complete **three-tier architecture** on AWS consisting of:

- **Web Tier** – Publicly accessible EC2 instances behind an **External Application Load Balancer**, serving the front-end and forwarding API requests to the application tier.
- **Application Tier** – EC2 instances hosted in private subnets and fronted by an **Internal Application Load Balancer**, processing business logic and API requests.
- **Database Tier** – A **MySQL Amazon RDS** instance running in private subnets, accessible only from the Application tier for enhanced security.

All layers are logically isolated using **custom VPC subnets** and protected with **security groups** following the least-privilege principle.

Connectivity has been validated end-to-end, and the Web Server successfully retrieves data from the database through the Application Server, confirming that all tiers are working together correctly.

The following screenshots confirm that the application is fully functional. The web tier successfully renders the front-end UI and retrieves data from the application and database tiers, verifying end-to-end connectivity across the entire three-tier architecture.

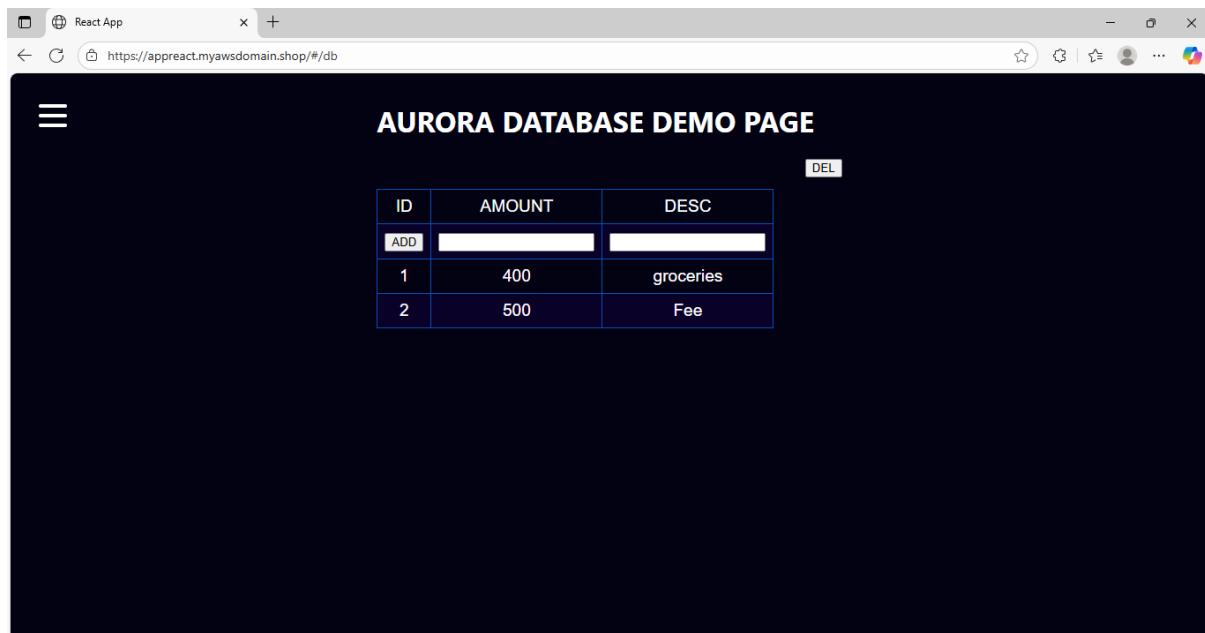
React App

https://appreact.myawsdomain.shop/#/db

AURORA DATABASE DEMO PAGE

ID	AMOUNT	DESC
ADD		
1	400	groceries
2	500	Fee

DEL



```
ec2-user@ip-172-16-3-247:~$ curl http://localhost:4000/transaction
{"result":[{"id":1,"amount":400,"description":"groceries"}, {"id":2,"amount":500,"description":"Fee"}]}[ec2-user@ip-172-16-3-247 ~]$
```

