

# AWS 3-Tier Architecture Implementation Guide

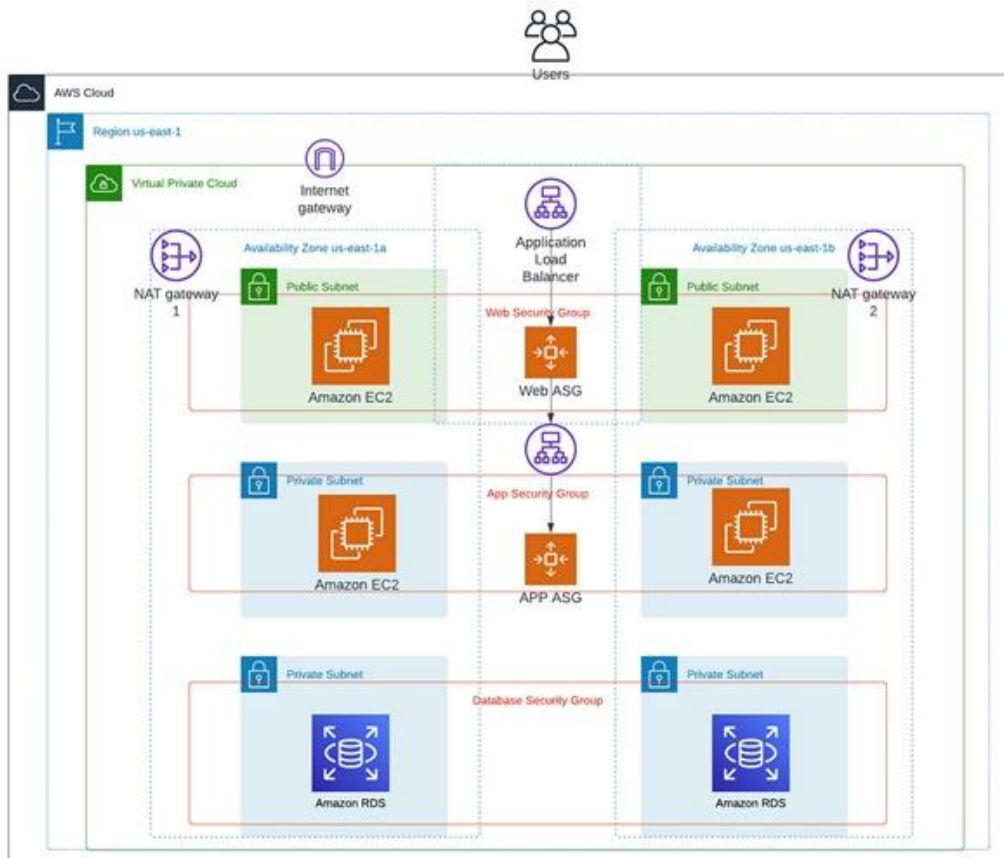
**Project:** Scalable 3-Tier Web Application on AWS

**Author:** Bhargav Mandapati

## Project Overview

This document provides a step-by-step implementation guide for building a 3-tier web application architecture on AWS, including all necessary screenshots and configurations.

## Architecture Diagram

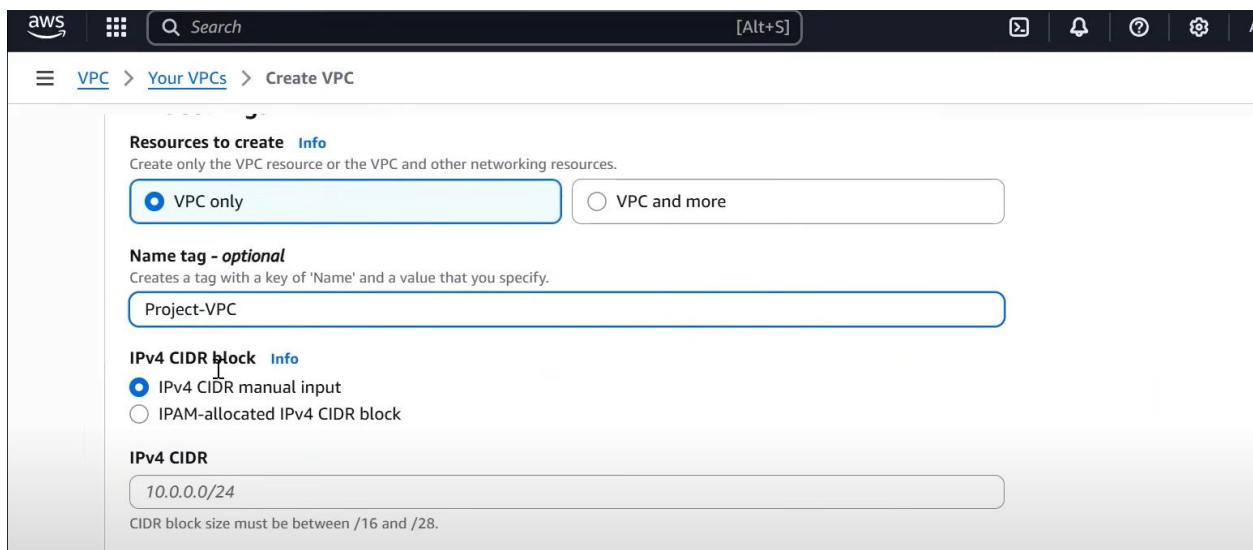


---

## Step 1: VPC Setup

### 1.1 Create VPC

1. Navigate to VPC Dashboard in AWS Console
2. Click "Create VPC"
3. Configure VPC settings:
  - Name: Project-VPC
  - Tenancy: Default



The screenshot shows the AWS Management Console 'Create VPC' page. The breadcrumb navigation at the top reads 'VPC > Your VPCs > Create VPC'. The main content area is titled 'Resources to create' with an 'Info' link. Below the title, it says 'Create only the VPC resource or the VPC and other networking resources.' There are two radio button options: 'VPC only' (which is selected) and 'VPC and more'. Below this is the 'Name tag - optional' section, which states 'Creates a tag with a key of 'Name' and a value that you specify.' The text 'Project-VPC' is entered in the input field. The next section is 'IPv4 CIDR block' with an 'Info' link. It has two radio button options: 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'. The 'IPv4 CIDR' input field contains '10.0.0.0/24'. A note at the bottom states 'CIDR block size must be between /16 and /28.'

### 1.2 Create Internet Gateway

1. Go to Internet Gateways in VPC Dashboard
2. Click "Create Internet Gateway"
3. Name: Project-IGW
4. Attach to VPC: Project-VPC

### Internet gateway settings

**Name tag**  
Creates a tag with a key of 'Name' and a value that you specify.

**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	
<input type="text" value="Name"/>	<input type="text" value="Project-IGW"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

## 1.3 Create Subnets

### Public Subnet

- Public Subnet 1:**
  - Name: Public-Subnet-1
  - VPC: Project-VPC
  - Availability Zone: us-east-1a
- Public Subnet 2:**
  - Name: Public-Subnet-2
  - VPC: Project-VPC
  - Availability Zone: us-east-1b

### 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

### Private Subnets (App Tier)

- Private Subnet 1:**
  - Name: Private-Subnet-1
  - VPC: Project-VPC

#### Private Subnet 2:

- Name: Private-Subnet-2
- VPC: Project-VPC

### Creating subnets...

We are currently creating subnets.

Creating subnet 3

33%

#### Details

- ✓ Creating subnet 1
- ✓ Creating subnet 2
- Creating subnet 3
- Creating subnet 4
- Creating subnet 5
- Creating subnet 6

### Edit subnet associations

Change which subnets are associated with this route table.

#### Available subnets (2/6)

Filter subnet associations

<input type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	Private-Subnet-1	<a href="#">subnet-01564e42e8f8ec72a</a>	20.0.3.0/26	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input type="checkbox"/>	private-Subnet-4	<a href="#">subnet-0174170dbbd031944</a>	20.0.6.0/23	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input type="checkbox"/>	Private-Subnet-2	<a href="#">subnet-033ed59f10e9f0b8b</a>	20.0.4.0/25	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input checked="" type="checkbox"/>	Public-Subnet-2	<a href="#">subnet-053b9a5bef06665cb</a>	20.0.2.0/27	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input type="checkbox"/>	Private-Subnet-3	<a href="#">subnet-01c2b7a4b94fe92eb</a>	20.0.69.0/24	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input checked="" type="checkbox"/>	Public-Subnet-1	<a href="#">subnet-09e89e0c82fc29bfb</a>	20.0.1.0/28	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>

#### Selected subnets

## Database Subnets

- Private-Subnet-3:**
  - Name: Private-Subnet-3
  - VPC: Project-VPC
- Private-Subnet-4:**
  - Name: Private-Subnet-4
  - VPC: Project-VPC

### Edit subnet associations

Change which subnets are associated with this route table.

#### Available subnets (4/6)

Filter subnet associations

<input type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	Private-Subnet-1	<a href="#">subnet-01564e42e8f8ec72a</a>	20.0.3.0/26	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input checked="" type="checkbox"/>	private-Subnet-4	<a href="#">subnet-0174170dbbd031944</a>	20.0.6.0/23	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input checked="" type="checkbox"/>	Private-Subnet-2	<a href="#">subnet-033ed59f10e9f0b8b</a>	20.0.4.0/25	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input type="checkbox"/>	Public-Subnet-2	<a href="#">subnet-053b9a5bef06665cb</a>	20.0.2.0/27	-	<a href="#">rtb-0e672ecb1a2e90bda / Public</a>
<input checked="" type="checkbox"/>	Private-Subnet-3	<a href="#">subnet-01c2b7a4b94fe92eb</a>	20.0.69.0/24	-	<a href="#">Main (rtb-01b1856bf90f3fbda)</a>
<input type="checkbox"/>	Public-Subnet-1	<a href="#">subnet-09e89e0c82fc29bfb</a>	20.0.1.0/28	-	<a href="#">rtb-0e672ecb1a2e90bda / Public</a>

**Create NAT gateway** [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

**NAT gateway settings**

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

Project-NAT

The name can be up to 256 characters long.

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

subnet-09e89e0c82fc29bfb (Public-Subnet-1)

**Connectivity type**  
Select a connectivity type for the NAT gateway.

☒ Public  
☐ Private

## 1.4 Project-NAT

1. Go to NAT Gateways in VPC Dashboard
2. Click "Project-NAT"
3. Configure:
  - o Name: 3-tier-nat-gateway
  - o Subnet: Public-Subnet-1
  - o Elastic IP: Allocate new

✓ Elastic IP address 18.141.134.126 (eipalloc-03bbb0a392eac911f) allocated.

**Elastic IP allocation ID** [Info](#)

Assign an Elastic IP address to the NAT gateway.

eipalloc-03bbb0a392eac911f [Allocate Elastic IP](#)

► **Additional settings** [Info](#)

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

Key	Value - optional	
Q Name	Q Project-NAT	<a href="#">Remove</a>

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create NAT gateway](#)

## 1.5 Configure Route Tables

### Public Route Table

1. Create Route Table:
  - o Name: Public-Route-Table
  - o VPC: Project-VPC
2. Add Routes:
  - o Destination: 0.0.0.0/0
  - o Target: Internet Gateway
3. Associate with Public Subnet

## Edit routes

Destination	Target	Status
20.0.0.0/16	local	✓ Active
<input type="text" value="Q 0.0.0.0/0"/>	<input type="text" value="local"/>	
	NAT Gateway	-
	<input type="text" value="Q nat-0a22bd989986eba9a"/>	




[Add route](#)

## Private Route Table

1. Create Route Table:
  - Name: Private-Route-Table
  - VPC: Project-VPC
2. Add Routes:
  - Destination: 0.0.0.0/0
  - Target: NAT Gateway
3. Associate with Private Subnets

## Database Route Table

1. Create Route Table:
  - Name: Database-Route-Table
  - VPC: Project-VPC
2. Associate with Database Subnets (no internet access)

aws  [Alt+S]   

☰ [VPC](#) > [Route tables](#) > Create route table

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

**VPC**  
The VPC to use for this route table.

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

**Key** **Value - optional**

[Add new tag](#)

You can add 49 more tags.

### Create route table Info

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

#### Route table settings

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

**VPC**  
The VPC to use for this route table.

## Step 2: Security Groups Configuration

### 2.1 Web Tier Security Group

- Create Security Group:
  - Name: Web-Tier-SG
  - Description: Security group for web servers
  - VPC: Project-VPC
- Inbound Rules:
  - HTTP (80): Source 0.0.0.0/0
  - HTTPS (443): Source 0.0.0.0/0
  - SSH (22): Source Your IP

### Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	
sgr-0436fab193c7134f2	SSH	TCP	22	Cu...		Delete
-	HTTP	TCP	80	An...	0.0.0.0/0	Delete
-	MYSQL/Aurora	TCP	3306	An...	0.0.0.0/0	Delete

Add rule

### 2.2 App Tier Security Group

- Create Security Group:
  - Name: App-Tier-SG
  - Description: Security group for application servers
  - VPC: Project-VPC
- Inbound Rules:
  - HTTP (80): Source Web-Tier-SG
  - SSH (22): Source Web-Tier-SG

### 2.3 Database Tier Security Group

1. Create Security Group:
  - Name: Database-Tier-SG
  - Description: Security group for database
  - VPC: Project-VPC
2. Inbound Rules:
  - MySQL (3306): Source App-Tier-SG

The screenshot shows the AWS Management Console interface for creating a DB subnet group. The breadcrumb navigation at the top reads: [Aurora and RDS](#) > [Subnet groups](#) > [Create DB subnet group](#). The main heading is **Create DB subnet group**. Below it, a subheading states: "To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC." The form is titled **Subnet group details** and contains three sections: **Name** with a text input field containing "Project-Subnet-Group" and a note "You won't be able to modify the name after your subnet group has been created."; **Description** with an empty text input field; and **VPC** with a dropdown menu showing "Choose a VPC".

## Step 3: Database Setup (RDS)

### 3.1 Create Database Subnet Group

1. Navigate to RDS Dashboard
2. Go to Subnet Groups
3. Click "Create DB Subnet Group"
4. Configure:
  - Name: database-subnet-group
  - Description: Subnet group for 3-tier database
  - VPC: Project-VPC
  - Subnets: Select both database subnets

### 3.2 Create RDS Instance

1. Click "Create Database"
2. Engine: MySQL
3. Template: Free tier
4. Settings:
  - DB Instance Identifier: project-database



- 
- Master Username: admin
- Master Password: [Your Password]
- 5. Instance Configuration:
  - DB Instance Class: db.t3.micro
- 6. Connectivity:
  - VPC: Project-VPC
  - Subnet Group: database-subnet-group
  - Security Group: Database-Tier-SG
  - No public access

☰ [Aurora and RDS](#) > Create database

**Create database** [Info](#)

---

**Choose a database creation method**

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

☐ **Easy create**  
 Use recommended best-practice configurations. Some config after the database is created.

---

**Engine options**

Engine type [Info](#)

☐ Aurora (MySQL Compatible)

☒ **Aurora (PostgreSQL Compatible)**

☐ MySQL

☐ PostgreSQL

☐ MariaDB

☐ Oracle

☰ [Aurora and RDS](#) > Create database

on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

☐ **Multi-AZ DB cluster deployment (3 instances)**  
 Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
 

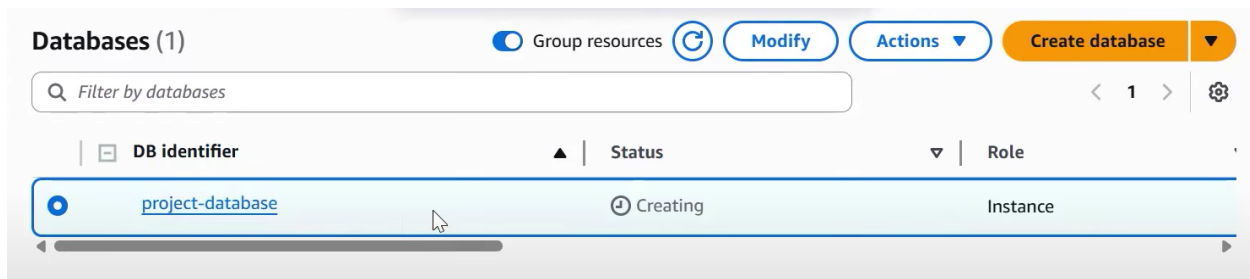
- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

☒ **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:
 

- 99.95% uptime
- Redundancy across Availability Zones

☐ **Single-AZ DB instance deployment (1 instance)**  
 Creates a single DB instance without standby instances. This setup provides:
 

- 99.5% uptime
- No data redundancy



```
root@ip-20-0-1-9:~# sudo systemctl start mysql.service
root@ip-20-0-1-9:~# mysql -h project-database.crkk88wuq912.ap-southeast-1.rds.amazonaws.com -u admin -p
Enter password: |
```

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

mysql> create database devops;
Query OK, 1 row affected (0.01 sec)

mysql> |
```

```
mysql> use devops
Database changed
mysql> clear
mysql> CREATE TABLE Details(
-> ID int NOT NULL,
-> LastName varchar(255) NOT NULL,
-> FirstName varchar(255),
-> Age int,
-> PRIMARY KEY(ID)
-> );
Query OK, 0 rows affected (0.03 sec)

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

mysql> INSERT INTO Details(ID, LastName, FirstName, Age)
-> VALUES (405, 'Teja', 'Ravi', 22);
Query OK, 1 row affected (0.01 sec)

mysql>
```

---

## Step 4: Application Load Balancer Setup

### 4.1 Create Application Load Balancer

1. Navigate to EC2 Dashboard → Load Balancers
2. Click "Create Load Balancer"
3. Select "Application Load Balancer"
4. Configure:
  - Name: Public-Load-Balancer
  - Scheme: Internet-facing
  - IP Address Type: IPv4
  - VPC: Project-VPC
  - Subnets: Select both public subnets
  - Security Group: Web-Tier-SG

## 4.2 Create Target Group

1. Create Target Group:
  - Name: Web-Tier-TG
  - Protocol: HTTP
  - Port: 80
  - VPC: Project-VPC

---

## Step 5: Launch The Instances

```
root@ip-20-0-2-12:~  
[root@ip-20-0-2-12 ~]# yum install nginx -y  
Last metadata expiration check: 0:00:43 ago on Tue Jun 10 16:25:18 2025.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository
Installing:			
nginx	x86_64	1:1.26.3-1.amzn2023.0.1	amazonlinu
Installing dependencies:			
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinu
gperftools-libs	x86_64	2.9.1-1.amzn2023.0.3	amazonlinu
libunwind	x86_64	1.4.0-5.amzn2023.0.2	amazonlinu
nginx-core	x86_64	1:1.26.3-1.amzn2023.0.1	amazonlinu
nginx-filesystem	noarch	1:1.26.3-1.amzn2023.0.1	amazonlinu
nginx-mimetypes	noarch	2.1.49-3.amzn2023.0.3	amazonlinu

```
=====
```

Transaction Summary

Install 7 Packages

Total download size: 1.1 M  
Installed size: 3.6 M  
Downloading Packages:  
(1/7): generic-logos-httpd-18.0.0-12 0% [ ] --- B/s | 0

### Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	
sgr-0436fab193c7134f2	SSH	TCP	22	Cu...		Delete
-	HTTP	TCP	80	An...	0.0.0.0/0	Delete
-	MYSQL/Aurora	TCP	3306	An...	0.0.0.0/0	Delete

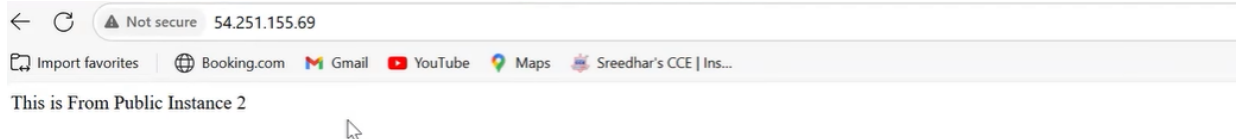
Add rule

← ↻ ⚠ Not secure 54.151.179.131

Import favorites | Booking.com | Gmail | YouTube | Maps | Sreedhar's CCE | Ins...

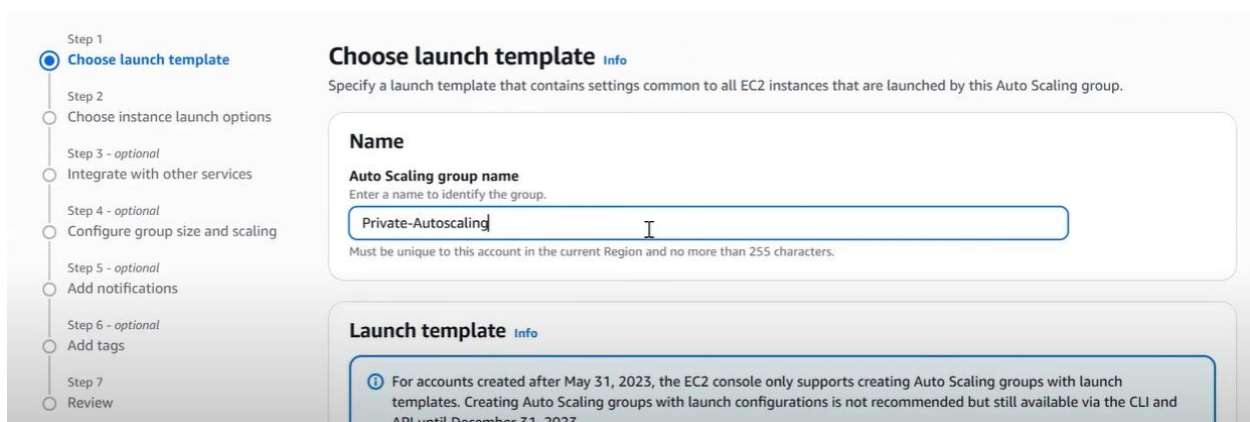
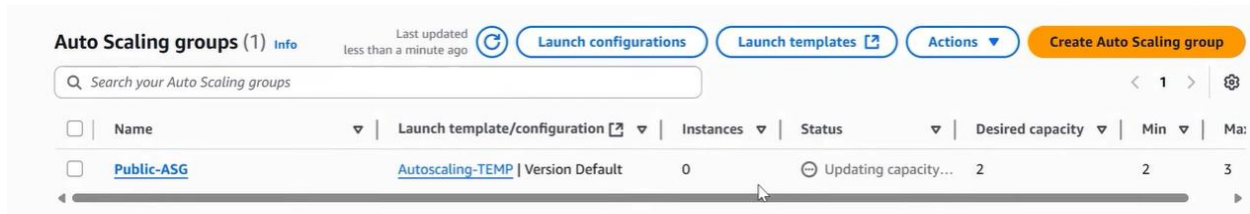
This is From Instance 1





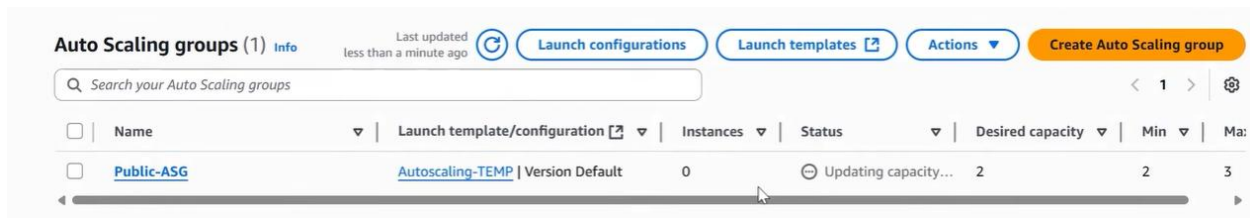
## 5.1 Create Launch Template

1. Navigate to EC2 Dashboard → Launch Templates
2. Click "Create Launch Template"
3. Configure:
  - Name: Web-Tier-Launch-Template
  - AMI: Amazon Linux 2
  - Instance Type: t2.micro
  - Key Pair: Select your key pair
  - Security Group: Public-ASG



## 5.2 Create Auto Scaling Group

1. Navigate to Auto Scaling Groups
2. Click "Create Auto Scaling Group"
3. Configure:
  - Name: Public-ASG
  - Launch Template: Web-Tier-Launch-Template
  - VPC: Project-VPC
  - Subnets: Select both public subnets
  - Load Balancer: Public-Load-Balancer
  - Target Group: Web-Tier-TG
  - Desired Capacity: 2
  - Min Capacity: 2
  - Max Capacity: 4



## Step 6: Application Tier Setup

### 6.1 Create App Tier Launch Template

1. Create Launch Template:
  - Name: App-Tier-Launch-Template
  - AMI: Amazon Linux 2
  - Instance Type: t2.micro
  - Key Pair: Select your key pair
  - Security Group: App-Tier-SG
  - Subnet: Private subnets
  - User Data Script:

### 6.2 Create App Tier Auto Scaling Group

1. Configure Auto Scaling Group:
  - Name: App-Tier-ASG
  - Launch Template: App-Tier-Launch-Template
  - Subnets: Select both private subnets
  - Desired Capacity: 2

- Min Capacity: 2
  - Max Capacity: 4
- 

## Project Summary

### What We Built:

- VPC with public, private, and database subnets
- Internet Gateway and NAT Gateway
- Security Groups for each tier
- Application Load Balancer
- Auto Scaling Groups for web and app tiers
- RDS MySQL database
- CloudWatch monitoring

### Key Achievements:

- High availability across multiple AZs
  - Auto scaling based on demand
  - Secure network architecture
  - Cost-effective resource utilization
  - Comprehensive monitoring
-