

CAPSTONE PROJECT 1

Project Report: Amazon EC2 Auto Scaling with RDS Integration

Submitted by: Vikram

Description

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using EC2 eliminates the need to invest in hardware upfront, allowing faster development and deployment of applications.

With EC2, users can launch as many or as few virtual servers as needed, configure networking and security, and manage storage resources effectively. EC2 also enables automatic scaling based on demand, improving cost efficiency and handling spikes in traffic effortlessly.

Problem Statement

Company ABC wants to migrate their existing product infrastructure to AWS. The product consists of:

- A MySQL Database
- A PHP-based Website

The company requires **high availability** for its application. To achieve this, the website should be deployed on **multiple EC2 instances** behind an **Auto Scaling group**, and the database should be hosted using **Amazon RDS (MySQL)**.

Steps to Solve

Step 1: Create RDS MySQL Database

1. Go to AWS Console → RDS → Create Database → MySQL
 2. Choose:
 - DB Name: intel
 - Master Username: intel
 - Master Password: intel123
 3. Select **Free tier** template (db.t3.micro)
 4. Disable **public access** for security
 5. Add inbound rule in **RDS Security Group**:
 - Type: MySQL/Aurora (3306)
 - Source: EC2 Security Group
-

Create database Info

Free plan has access to limited features and resources

The free plan limits the features and resources that are available for RDS and Aurora databases. Upgrade your account plan to remove all limitations. [Learn more](#)[Upgrade plan](#)

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 configuration options can be changed after the database is created.

Engine options

Engine type Info Aurora (MySQL Compatible) Aurora (PostgreSQL Compatible) MySQL[View details](#)[View details](#)[View details](#)[View details](#)[View details](#)[View details](#)[View details](#)[View details](#)[View details](#)

▼ Credentials Settings

Master username InfoType a login ID for the master user of your DB instance.

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 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.

 Self managed

Create your own password or have RDS create a password that you manage.

 Auto generate password

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

Master password Info

Aurora and RDS

<

intel

Actions

board

bases
performance insights
shots
RTS in Amazon S3
managed backups
viewed instances
esevent groups
meter groups
join groups
join engine versions
ETL integrationsts
t subscriptions

Summary

DB identifier

intel

Status

Backing-up

Role

Instance

Engine

MySQL Community

CPU

8.30%

Class

db.t4g.micro

Current activity

0 Connections

Recommendations

Region & AZ

us-east-1b

Connectivity & security

Endpoint & port

Endpoint
 intel.cazam602g56y.us-east-1.rds.amazonaws.comPort
3306

Networking

Availability Zone
us-east-1bVPC
default vpc (vpc-03af9fa3d1eb0c8bf)
Subnet group

Security

VPC security groups
mysql database sg (sg-025d19122d084b06b)Active
Publicly accessible
No

Step 2: Launch an EC2 Instance

- Navigate to **AWS Management Console** → **EC2** → **Launch Instance**
- Choose:
 - **Ubuntu**
 - **T3.micro** (Free Tier)
- Configure:
 - Enable **Auto-assign Public IP**
 - Add Tag: Name = EC2-WebServer
- Create a **Security Group**:
 - Allow **All Traffic** → Anywhere (0.0.0.0/0)

Launch and connect via EC2 Instance Connect

The screenshot shows the AWS EC2 Instances page. At the top, there's a header with 'Instances' and various navigation icons. Below it, a search bar says 'Find Instance by attribute or tag (case-sensitive)' and a status filter says 'Running'. The main table has columns for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', 'Alarm status', 'Availability Zone', and 'Public IPv4'. One instance is listed: 'project' (Instance ID: i-0fcd2533f4a91ec3a), which is 'Running' (status check: 3/3 checks passed), in 't3.micro' type, 'us-east-1b' availability zone, and has a public IP of 'ec2-54-88-6'. There are also 'View alarms' and 'Launch instances' buttons.

Step 3: Install Apache, PHP, and MySQL Client

Run the following commands in your terminal:

```
sudo apt update
```

```
sudo apt install apache2
```

```
cd /var/www/html
```

Copy the **Public IP** of your EC2 instance.

Paste it into your web browser →

You should see your Apache default page.

Come back to terminal And Run these commands

```
sudo rm -r index.html
```

```
sudo nano index.php
```

```
# (Paste your PHP code here)
# Change the endpoint in the PHP code to your RDS endpoint
# (Press Ctrl + S to save and Ctrl + X to exit)
```

```
ubuntu@ip-10-0-3-162:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1541 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [292 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.4 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1496 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [301 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.3 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2170 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1260 kB]
```

```
59 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-10-0-3-162:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libaprilt64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 55 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8090 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libaprilt64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.8 [1331 kB]
```

```
ubuntu@ip-10-0-3-162:~$ cd /var/www/html
ubuntu@ip-10-0-3-162:/var/www/html$
```

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.Load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
```

```
ubuntu@ip-10-0-3-162:/var/www/html$ sudo rm -r index.html
ubuntu@ip-10-0-3-162:/var/www/html$ sudo nano index.php
```

Now install PHP and MySQL tools:

```
sudo add-apt-repository ppa:ondrej/php
```

```
sudo apt install php5.6
```

```
sudo apt install php5.6-mysqli
```

```
sudo apt install mysql-client
```

```
ubuntu@ip-10-0-3-162:/var/www/html$ sudo add-apt-repository ppa:ondrej/php
PPA publishes dbgsym, you may need to include 'main/debug' component
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ondrej/php/ubuntu/
Suites: noble
Components: main
'
Description:
Co-installable PHP versions: PHP 5.6, PHP 7.x, PHP 8.x and most requested extensions are included. Packages are provided for *Current* Ubuntu *LTS* releases (https://www.ubuntu.com/Releases). Expanded Security Maintenance releases ARE NOT supported.

Debian stable, oldstable and Debian LTS packages are provided from a separate repository: https://deb.sury.org/#debian-dpa

You can get more information about the packages at https://deb.sury.org

BUGS&FEATURES: This PPA has a issue tracker:
https://deb.sury.org/#bug-reporting

Issues reported in a private email don't scale and most likely will be ignored. I simply don't have capacity to answer questions privately.

CAVEATS:
1. If you are using apache2, you are advised to add ppa:ondrej/apache2
2. If you are using nginx, you are advised to add ppa:ondrej/nginx
```

```
Reading packages...
ubuntu@ip-10-0-3-162:/var/www/html$ sudo apt install php5.6
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php5.6 libpcre3 php-common php5.6-cli php5.6-common php5.6-json php5.6-opcache php5.6-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php5.6 libpcre3 php-common php5.6-cli php5.6-common php5.6-json php5.6-opcache php5.6-readline
0 upgraded, 9 newly installed, 0 to remove and 55 not upgraded.
Need to get 3883 kB of archives.
After this operation, 14.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libpcre3 amd64 2:8.39-15build1 [248 kB]
Get:2 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php-common all 2:96+ubuntu24.04.1+deb.sury.org+1 [13.3 kB]
Get:3 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-common amd64 5.6.40-86+ubuntu24.04.1+deb.sury.org+1 [677 kB]
Get:4 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-json amd64 5.6.40-86+ubuntu24.04.1+deb.sury.org+1 [19.0 kB]
Get:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-opcache amd64 5.6.40-86+ubuntu24.04.1+deb.sury.org+1 [66.7 kB]
Get:6 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-readline amd64 5.6.40-86+ubuntu24.04.1+deb.sury.org+1 [13.6 kB]
```

```
Reading packages...
ubuntu@ip-10-0-3-162:/var/www/html$ sudo apt install php5.6-mysqli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php5.6-mysql' instead of 'php5.6-mysqli'
The following NEW packages will be installed:
  php5.6-mysql
0 upgraded, 1 newly installed, 0 to remove and 55 not upgraded.
Need to get 149 kB of archives.
After this operation, 560 kB of additional disk space will be used.
Get:1 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-mysql amd64 5.6.40-86+ubuntu24.04.1+deb.sury.org+1 [149 kB]
Fetched 149 kB in 1s (200 kB/s)
Selecting previously unselected package php5.6-mysql.
(Reading database ... 72590 files and directories currently installed.)
Preparing to unpack .../php5.6-mysql_5.6.40-86+ubuntu24.04.1+deb.sury.org+1_amd64.deb ...
Unpacking php5.6-mysql (5.6.40-86+ubuntu24.04.1+deb.sury.org+1) ...
Setting up php5.6-mysql (5.6.40-86+ubuntu24.04.1+deb.sury.org+1) ...
```

```

ubuntu@ip-10-0-3-162:/var/www/html$ sudo apt install mysql-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  mysql-client-8.0 mysql-client-core-8.0 mysql-common
The following NEW packages will be installed:
  mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common
0 upgraded, 4 newly installed, 0 to remove and 55 not upgraded.
Need to get 2779 kB of archives.
After this operation, 61.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [2740 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.8+1.1.0build1 [6746 B]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.43-0ubuntu0.24.04.2 [22.4 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.43-0ubuntu0.24.04.2 [9404 B]
Fetched 2779 kB in 0s (51.2 MB/s)
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 72605 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.43-0ubuntu0.24.04.2_amd64.deb ...
Unpacking mysql-client-core-8.0 (8.0.43-0ubuntu0.24.04.2) ...
Selecting previously unselected package mysql-common.
Preparing to unpack .../mysql-common_5.8+1.1.0build1_all.deb ...
Unpacking mysql-common (5.8+1.1.0build1)

```

Step 4: Verify Apache Server

1. Copy the **Public IP** of your EC2 instance.
2. Paste it into your web browser →
You should see your PHP form (instead of the Apache default page).



Step 5: Connect to RDS MySQL Database from EC2

Run the following command:

```
sudo mysql -h <hostname> -u <username> -p<password>
```

Example:

```
sudo mysql -h your-rds-endpoint.rds.amazonaws.com -u intel -pintel123
```

- When prompted, it will connect to your RDS database.

```
ubuntu@ip-10-0-3-162:/var/www/html$ sudo mysql -h intel.cazam602g56y.us-east-1.rds.amazonaws.com -u intel -pintel123
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 32
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> |
```

Step 6: Database Setup Commands

After connecting to MySQL, execute these commands:

```
show databases;
```

```
create database intel;
```

```
use intel;
```

```
create table data(firstname varchar(50), email varchar(50));
```

```
select * from data;
```

```
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.01 sec)

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

mysql> use intel;
Database changed
mysql> create table data(firstname varchar(50), email varchar (50));
Query OK, 0 rows affected (0.05 sec)

mysql> select * from data;
Empty set (0.00 sec)
```

Step 7: Test the Web Application

1. Open the website in your browser using:
2. [http://<your-ec2-public-ip>/](http://<your-ec2-public-ip>)
3. Enter a **Name** and **Email**, then click **Submit**.
4. If the configuration is correct, you'll see:
5. [New record created successfully](#)
6. Go back to MySQL and run:
7. [select * from data;](#)

→ The inserted data should be visible in the `data` table.



Name:
VIKRAM

Email:
gangadharavikram1999@gmail.com

Submit

Name:
[empty input field]

Email:
[empty input field]

Submit

New record created successfully

```
mysql> select * from data;  
Empty set (0.00 sec)
```

```
mysql> select * from data;  
+-----+  
| firstname | email  
+-----+  
| VIKRAM | gangadharavikram1999@gmail.com |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> █
```

Step 8: Enable Auto Scaling

1. Create an **AMI** of your configured EC2 instance.
2. Go to **EC2 → Launch Template → Create Launch Template** using that AMI.
3. Create an **Auto Scaling Group**:
 - o Minimum instances: **2**
 - o Desired capacity: **2**
 - o Maximum instances: **4**
 - o Attach same **Security Group** and **subnets in multiple AZs**

Auto Scaling ensures high availability even if one instance fails.

The screenshot shows the 'Create New Stack' wizard in the AWS CloudFormation console. The current step is 'Set stack details'. The stack name is 'MyFirstStack'. The region is 'us-east-1'. The template type is 'CloudFormation template (.yaml or .json)'. The 'Next Step' button is visible at the bottom.

Stack details

Stack name
MyFirstStack

Region
us-east-1

Template type
CloudFormation template (.yaml or .json)

Next Step

Amazon Machine Images (AMIs) (1) Info					
Owned by me		<input type="text"/> Find AMI by attribute or tag			
<input type="checkbox"/>	Name 🔗	AMI name	AMI ID	Source	Owner
<input type="checkbox"/>	capstone image	ami-0cd4935c145a25f79	062250062838/capstone image	062250062838	Private

Select an AMI



▼ Application and OS Images (Amazon Machine Image) - required [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](#).

[Search our full catalog including 1000s of application and OS images](#)

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

[Owned by me](#)

[Shared with me](#)



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

capstone image
ami-0cd4935c145a25f79
2025-10-27T05:37:57.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description

-

Launch Templates (1) [Info](#)



[Actions](#) [Create launch template](#)

<input type="text"/> Search					
<input type="checkbox"/>	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time
<input type="checkbox"/>	lt-0baeecea6170a6d	demotemplate	1	1	2025-10-27T05:49:03.000Z

Options

Name**Auto Scaling group name**

Enter a name to identify the group.

project asg

Must be unique to this account in the current Region and no more than 255 characters.

Launch template

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.

demotemplate

[Create a launch template](#) **Version**

Default (1)

[Create a launch template version](#) **Description**[Launch template](#)[Instance type](#)

Ins

Group size

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.

aling

Units (number of instances)

**Desired capacity**

Specify your group size.

2

Scaling

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

4



Equal or greater than desired capacity

Automatic scaling - optional

project asg

project asg Capacity overview

arn:aws:autoscaling:us-east-1:062250062838:autoScalingGroup:fe9a36e9-1266-4575-9fbc-dcd3936e3c04:autoScalingGroupName/project asg

Desired capacity 2	Scaling limits (Min - Max) 2 - 4	Desired capacity type Units (number of instances)	Status Updating capacity
-----------------------	-------------------------------------	--	-----------------------------

Date created
Mon Oct 27 2025 11:24:00 GMT+0530 (India Standard Time)

Details Integrations Automatic scaling Instance management Instance refresh Activity Monitoring Tags - moved

Launch template

Launch template lt-00baecea6170a6d demotemplate	AMI ID ami-0cd4935c145a25f79	Instance type -	Owner arn:aws:iam::062250062838:root
Version Default	Security groups -	Security group IDs -	Create time Mon Oct 27 2025 11:19:03 GMT+0530 (India Standard Time)

✓ Final Output Verification

- **Website:** PHP form loads successfully on EC2
- **Database:** RDS connected and data inserted correctly
- **Auto Scaling:** Maintains minimum 2 active instances automatically
- **Security:** Proper inbound rules set for EC2 ↔ RDS

Conclusion

In this project, we successfully:

- Deployed a PHP-based website on an EC2 instance
- Configured Apache, PHP, and MySQL Client
- Integrated EC2 with RDS MySQL database
- Implemented Auto Scaling for high availability

This architecture provides a **scalable**, **fault-tolerant**, and **cost-efficient** cloud solution for Company ABC's web application on AWS.

