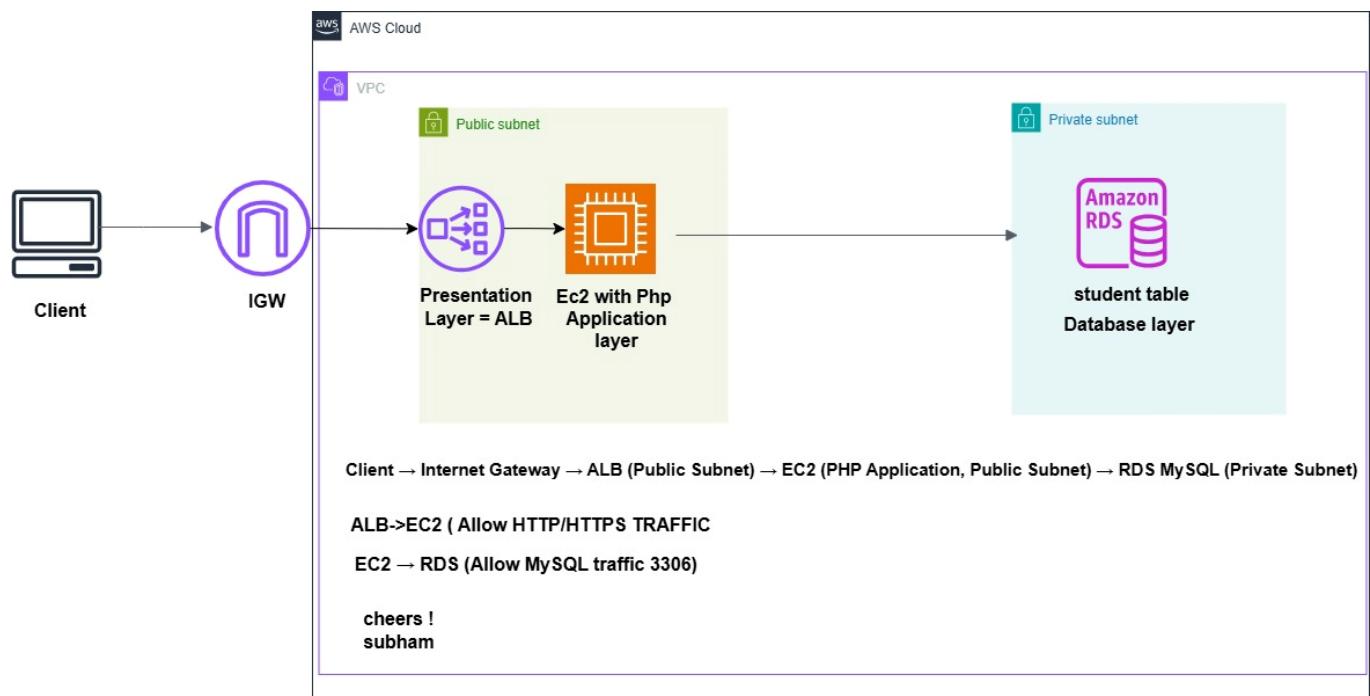




Project Name:

AWS 3-Tier Architecture with Load Balancer, PHP Application, and RDS MySQL



Step-by-Step Project Guide :

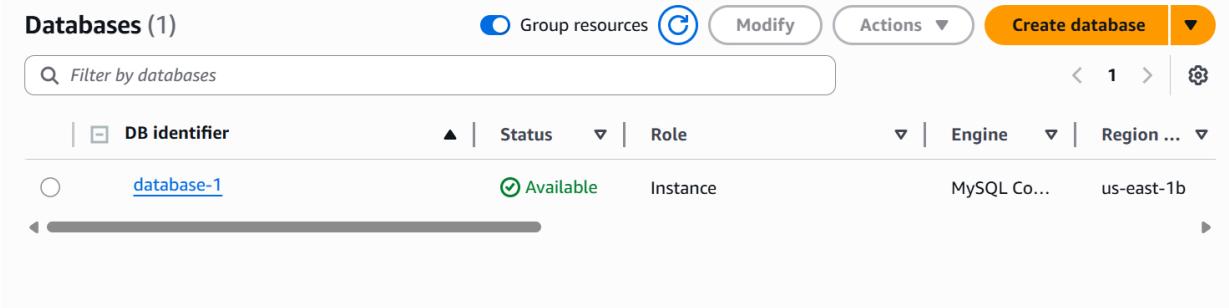
1) Networking & Security Groups

- RDS SG (sg-rds):** Inbound 3306 → Source = sg-ec2 (your EC2 security group).
- EC2 SG (sg-ec2):** Inbound 80 → Source = sg-alb (or 0.0.0.0/0 for testing). Inbound 22 from your own IP only.
- ALB SG (sg-alb):** Inbound 80 → Source = 0.0.0.0/0.
- RDS Public Accessibility = No.**
- EC2 must be in public subnet, RDS in private subnet.**

2) Create RDS MySQL

- Console → Create database → MySQL.

- DB subnet group → private subnets across AZs.
- Public accessibility → **No**.
- Master username: admin, Password: admin123.
- Initial DB name: student.
- Launch and note the **endpoint** (you gave it earlier):
- database-1.ckbei6u06u0d.us-east-1.rds.amazonaws.com



Databases (1)

Group resources **Modify** Actions ▾ Create database ▾

Filter by databases

DB identifier	Status	Role	Engine	Region ...
database-1	Available	Instance	MySQL Co...	us-east-1b

•

```
[ec2-user@ip-10-0-1-129 ~]$ mysql -h database-1.ckbei6u06u0d.us-east-1.rds.amazonaws.com -u admin -padmin123
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

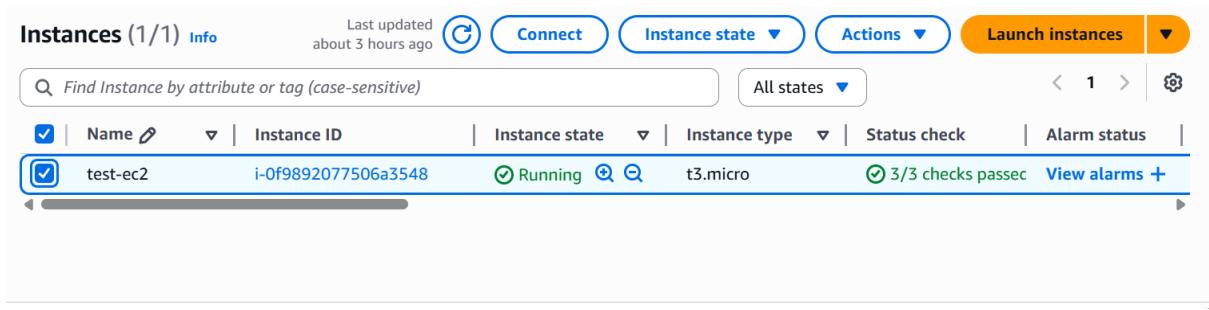
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [none]> ^C
```

•

3) Launch EC2 (public subnet)

- Amazon Linux 2023 or 2.
- Subnet = public, Auto-assign Public IP = Enabled.
- Security group = sg-ec2.
- Launch and connect (SSH or Session Manager).



```

# 
# ~\_\#\#\#
# ~\_\#\#\#\\
# ~\#\#
# ~\#/`-->
# ~`-'`/
# ~`-'`/`/
# /`m`'
#
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sat Sep 20 06:19:32 2025 from 18.206.107.28
[ec2-user@ip-10-0-1-129 ~]$ sudo yum update
Last metadata expiration check: 0:06:48 ago on Sat Sep 20 06:21:28 2025.
Dependencies resolved.
Nothing to do.
Complete!

```

4) Install Apache + PHP + MySQL client on EC2

Amazon Linux 2023:

```
sudo dnf update -y
```

```
sudo dnf install -y httpd php php-mysqlnd  
mariadb105
```

```
sudo systemctl enable --now httpd
```

(Amazon Linux 2 → use yum. Ubuntu → use
apt install apache2 php php-mysql mariadb-client.)

```
[ec2-user@ip-10-0-1-129 ~]$ sudo dnf install -y httpd php php-mysqlnd
Last metadata expiration check: 0:32:35 ago on Sat Sep 20 06:21:28 2025.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository | Size |
|=====
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.65-1.amzn2023.0.1	amazonlinux	47 k
php8_4	x86_64	8.4.10-1.amzn2023.0.1	amazonlinux	17 k
php8_4-mysqlnd	x86_64	8.4.10-1.amzn2023.0.1	amazonlinux	156 k
Installing dependencies:				
apr	x86_64	1.7.5-1.amzn2023.0.4	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httdp	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.65-1.amzn2023.0.1	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.65-1.amzn2023.0.1	amazonlinux	13 k
httpd-tools	x86_64	2.4.65-1.amzn2023.0.1	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux	176 k
libxml2	x86_64	1.1.43-1.amzn2023.0.2	amazonlinux	183 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k

```
i-0f9892077506a3548 (test-ec2)
PublicIPs: 18.212.9.134 PrivateIPs: 10.0.1.129
```

5) Create DB & table in RDS

On EC2:

```
mysql -h database-1.ckbei6u06u0d.us-east-1.rds.amazonaws.com -u admin -p
```

```
# enter: admin123
```

Inside MySQL:

```
CREATE DATABASE IF NOT EXISTS student;
```

```
USE student;
```

```
CREATE TABLE IF NOT EXISTS students (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(100),
```

```
    course VARCHAR(100)
```

```
);
```

```
INSERT INTO students (name, course) VALUES
```

```
('Rahul','B.Tech'),
```

```
('Priya','MBA'),
```

```
('Asha','BSc');
```

```
SELECT * FROM students;
```

```
MySQL [student]> SELECT * FROM students;
+----+-----+-----+
| id | name  | course |
+----+-----+-----+
| 1  | Rahul | B.Tech |
| 2  | Priya | MBA   |
| 3  | Asha  | BSc   |
+----+-----+-----+
3 rows in set (0.001 sec)
```

6) Create PHP app (hardcoded DB credentials)

On EC2:

```
sudo tee /var/www/html/index.php >
/dev/null <<'PHP'
<?php
$host = "database-1.ckbei6u06u0d.us-east-
1.rds.amazonaws.com";
$user = "admin";
$pass = "admin123";
```

```
$dbname = "student";  
  
$conn = new mysqli($host, $user, $pass,  
$dbname);  
  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn-  
    >connect_error);  
}  
  
  
  
$result = $conn->query("SELECT id, name,  
course FROM students");  
  
echo "<h2>Student Records</h2>";  
  
if ($result && $result->num_rows > 0) {  
    echo "<table border='1'  
cellpadding='6'><tr><th>ID</th><th>Name</t  
h><th>Course</th></tr>";  
  
    while($row = $result->fetch_assoc()) {
```

```

echo

"<tr><td>".htmlspecialchars($row['id'])."</td>
<td>".htmlspecialchars($row['name'])."</td><
td>".htmlspecialchars($row['course'])."</td><
/tr>";

}

echo "</table>";

} else {

echo "No records found";

}

$conn->close();

?>

```

PHP

```
[ec2-user@ip-10-0-1-129 ~]$ sudo tee /var/www/html/index.php > /dev/null <<'PHP'
<?php
    mysql -h database-1.ckbei6u06u0d.us-east-1.rds.amazonaws.com -u admin -padmin123
    sudo tee /var/www/html/index.php > /dev/null <<'PHP'

$host = "<RDS-ENDPOINT>";
$user = "<DB-USER>";
$pass = "<DB-PASS>";
$dbname = "student";

$conn = new mysqli($host, $user, $pass, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$result = $conn->query("SELECT id, name, course FROM students");
echo "<h2>Student Records</h2>";
if ($result->num_rows == 0) {

```

i-0f9892077506a3548 (test-ec2)

Public IPs: 18.212.9.134 Private IPs: 10.0.1.129

```
sudo chown -R apache:apache  
/var/www/html
```

```
sudo chmod -R 755 /var/www/html
```

```
sudo systemctl restart httpd
```

```
[ec2-user@ip-10-0-1-129 ~]$ sudo nano /var/www/html/index.php  
[ec2-user@ip-10-0-1-129 ~]$ sudo systemctl restart httpd
```

7) Local test on EC2

```
curl http://localhost/index.php
```

- ✓ If you see student table → it works.
- ✗ If error → check DB connection details & security groups.

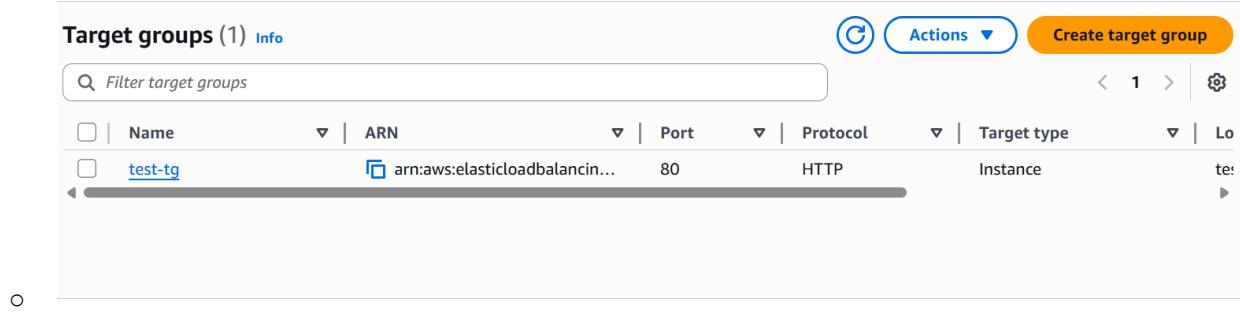
```
[ec2-user@ip-10-0-1-129 ~]$ curl http://localhost/index.php  
<h2>Student Records</h2><table border='1' cellpadding='6'><tr><th>ID</th><th>Name</th><th>Course</th></tr><tr><td>1</td><td>Rahul</td><td>B.Tech</td></tr><tr><td>2</td><td>Priya</td><td>MBA</td></tr><tr><td>3</td><td>Asha</td><td>B.Sc</td></tr></table>  
[ec2-user@ip-10-0-1-129 ~]$
```

i-0f9892077506a3548 (test-ec2) >
PublicIPs: 18.212.9.134 PrivateIPs: 10.0.1.129

8) Create Target Group & ALB

1. Target Group:

- Type = Instances
- Protocol = HTTP, Port = 80
- Health check path /index.php
- Register EC2.



The screenshot shows the AWS Lambda console's 'Target groups' section. It displays a single target group named 'test-tg'. The table includes columns for Name, ARN, Port, Protocol, Target type, and Location. The 'Name' column shows 'test-tg', the 'ARN' column shows 'arn:aws:elasticloadbalancing...', the 'Port' column shows '80', the 'Protocol' column shows 'HTTP', the 'Target type' column shows 'Instance', and the 'Location' column shows 'US East (N. Virginia)'. There are also 'Actions' and 'Create target group' buttons at the top right.

Name	ARN	Port	Protocol	Target type	Location
test-tg	arn:aws:elasticloadbalancing...	80	HTTP	Instance	US East (N. Virginia)

◦

2. ALB:

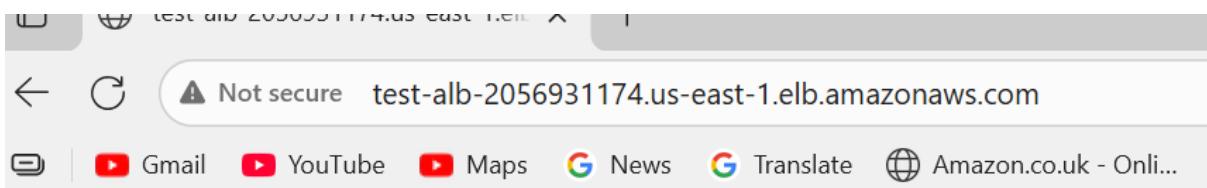
- Internet-facing, public subnets in 2 AZs.
- Security group = sg-alb (inbound 80).
- Listener HTTP:80 → forward to target group.

The screenshot shows the AWS Load Balancers console. At the top, there's a header with the title "Load balancers (1)" and a note: "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." Below the header is a search bar labeled "Filter load balancers". To the right of the search bar are buttons for "Actions" and "Create load balancer". A table lists one load balancer: "test-alb" (Active, application, Internet-facing, IPv4, VPC ID: vpc-00d44c). Below the table, it says "0 load balancers selected" and "Select a load balancer above."

9) Test from Browser

Open:

<http://test-alb-xxxxx.us-east-1.elb.amazonaws.com/>



Student Records

ID	Name	Course
1	Rahul	B.Tech
2	Priya	MBA
3	Asha	BSc

- ALB forwards → EC2 → PHP runs → connects to RDS → displays student table.
-

10) Troubleshooting Checklist

- `sudo systemctl status httpd` → Apache running?
- `curl http://localhost/index.php` → works locally?
- `mysql -h ... -u admin -p` → EC2 can reach RDS?
- Security groups set correctly?
- ALB Target group shows **healthy**?

THANK YOU

STAY CONNECT WITH

[www.linkedin.com/in/ subham-badatya](https://www.linkedin.com/in/subham-badatya)