**Project : Using AWS Elastic Beanstalk to Set Up RDS and Access It from an EC2 Instance**

**Obejctive :**

The objective of this project is to **deploy a sample web application using AWS Elastic Beanstalk** with an **integrated Amazon RDS (MySQL) database**, and **securely access the database from a separate EC2 instance** within the same VPC.

This project demonstrates how to:

* Use Elastic Beanstalk to launch scalable web applications.
* Provision and configure an RDS database as part of the application environment.
* Control network access using VPC and security groups.
* Connect to the RDS instance from an external EC2 instance using MySQL client tools.
* Optionally enhance the setup with secure credential management and database test scripts.

This hands-on experience strengthens your understanding of **AWS services integration, networking, and secure infrastructure deployment in the cloud.**

**Introduction :**

In modern cloud-based applications, it's essential to deploy scalable, maintainable, and secure infrastructure that integrates application and database layers. AWS Elastic Beanstalk simplifies the deployment and management of web applications by automatically handling capacity provisioning, load balancing, scaling, and application health monitoring.

This project focuses on deploying a sample web application using AWS Elastic Beanstalk with a connected RDS (MySQL) database, and accessing that database securely from a separate EC2 instance within the same Virtual Private Cloud (VPC). By doing this, we explore how to establish secure communication between application and database tiers using AWS services, manage database access through security groups, and optionally implement enhancements like credential security and monitoring.

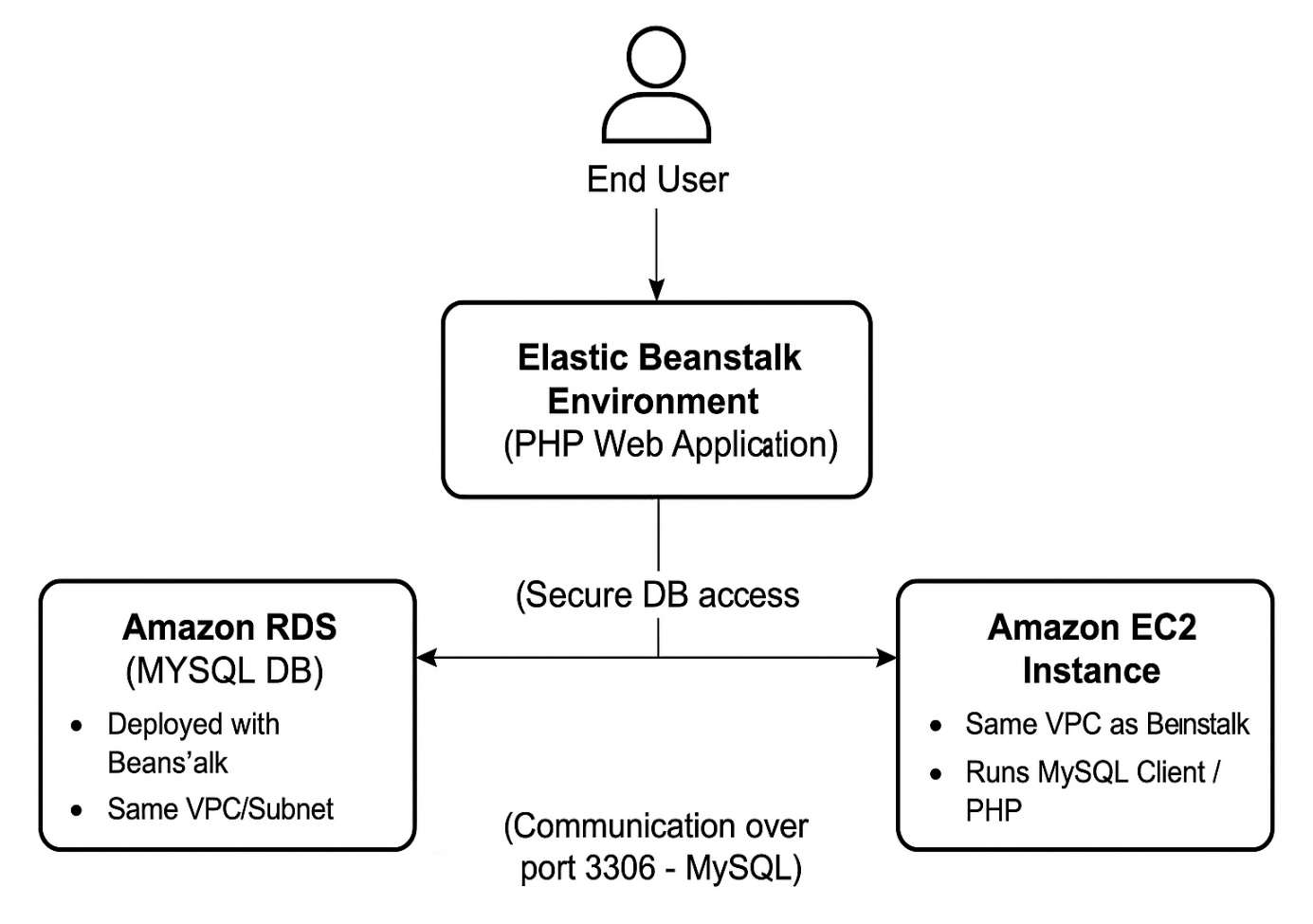
This project helps learners and cloud practitioners understand:

* Application deployment with Elastic Beanstalk
* Database provisioning with Amazon RDS
* Secure network communication within a VPC
* Real-world DevOps and cloud infrastructure integration best practices

**Techonology Stack :**

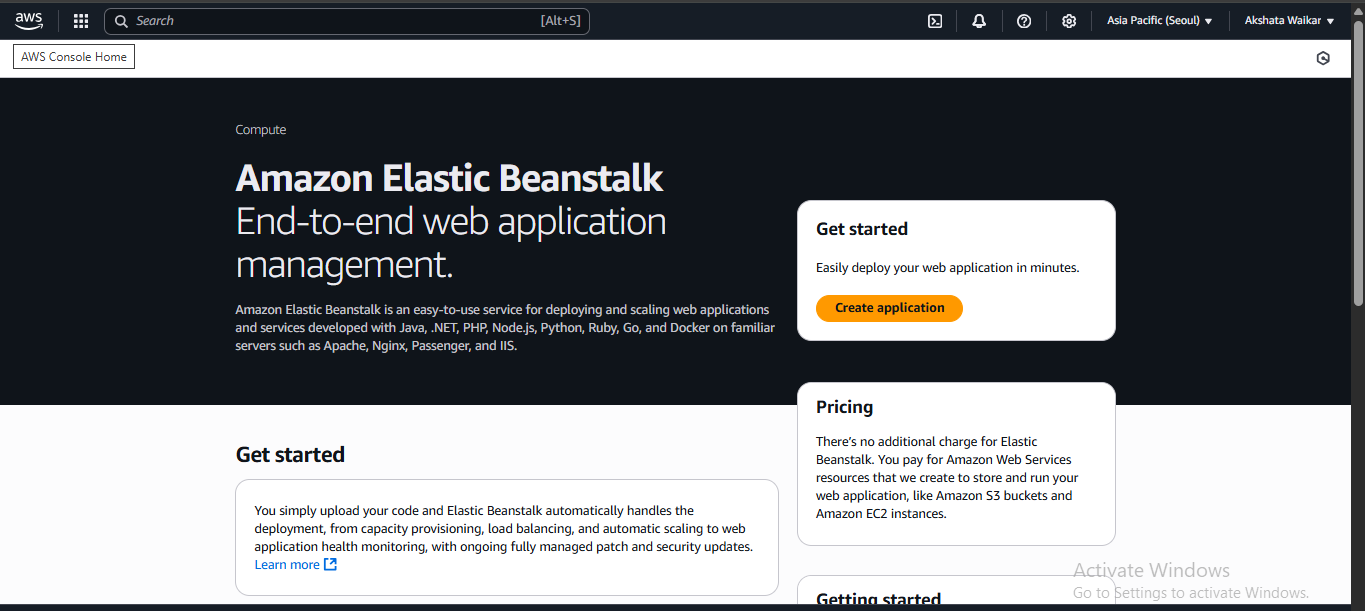
|  |  |  |
| --- | --- | --- |
| **Component** | **Technology / Tool** | **Purpose** |
| Application Platform | AWS Elastic Beanstalk (PHP Platform) | To deploy and manage the web application |
| Database | Amazon RDS (MySQL) | To store and manage relational data |
| Compute Instance | Amazon EC2 | To externally connect and interact with the RDS instance |
| Networking | Amazon VPC, Subnets, Security Groups | To ensure secure and isolated networking between services |

**System architecture diagram :**

****

**Implementation Steps :**

🔹 **Step 1: Open AWS Elastic Beanstalk Console**

****

Click **“Create application”**

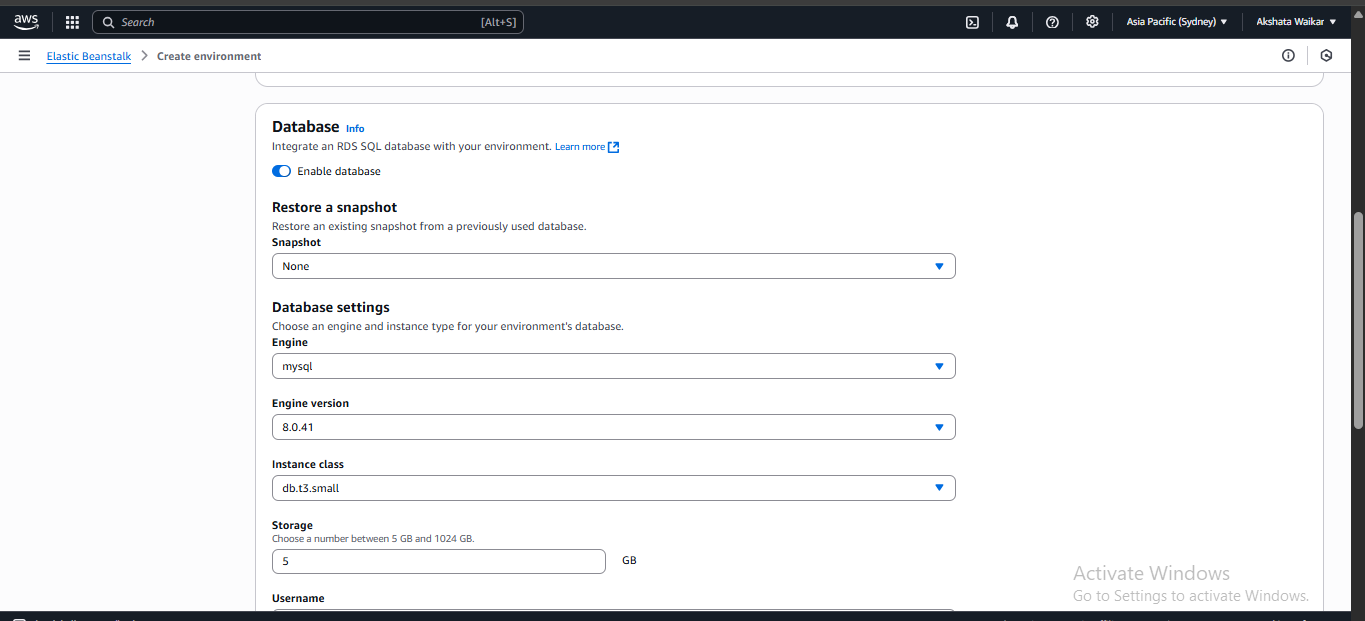
🔹 **Step 2: Configure Application**

- **Application name**: phpapp

- **Platform**: Use the latest PHP version (e.g., PHP 8.1)

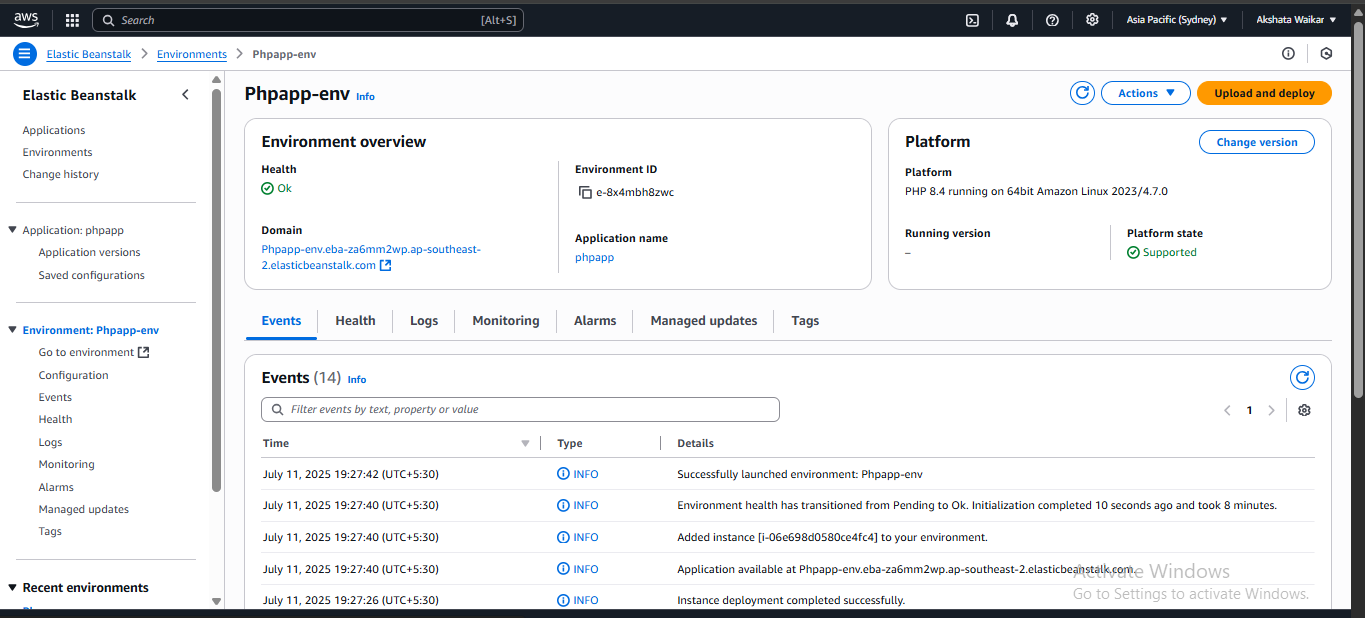
**🔹 Step 3: Add Amazon RDS Database**

1. Scroll to the **“Database”** section
2. Enable: ✅ **Create an Amazon RDS database with this environment**
3. Fill in details:
   * Engine: **MySQL**
   * Username: admin
   * Password: Akshata1307
   * DB instance class: db.t3.micro (Free Tier eligible)
   * Storage: 20 GB
   * Enable deletion protection: Optional
4. RDS will be deployed **in the same VPC/** **subnet** as Beanstalk.



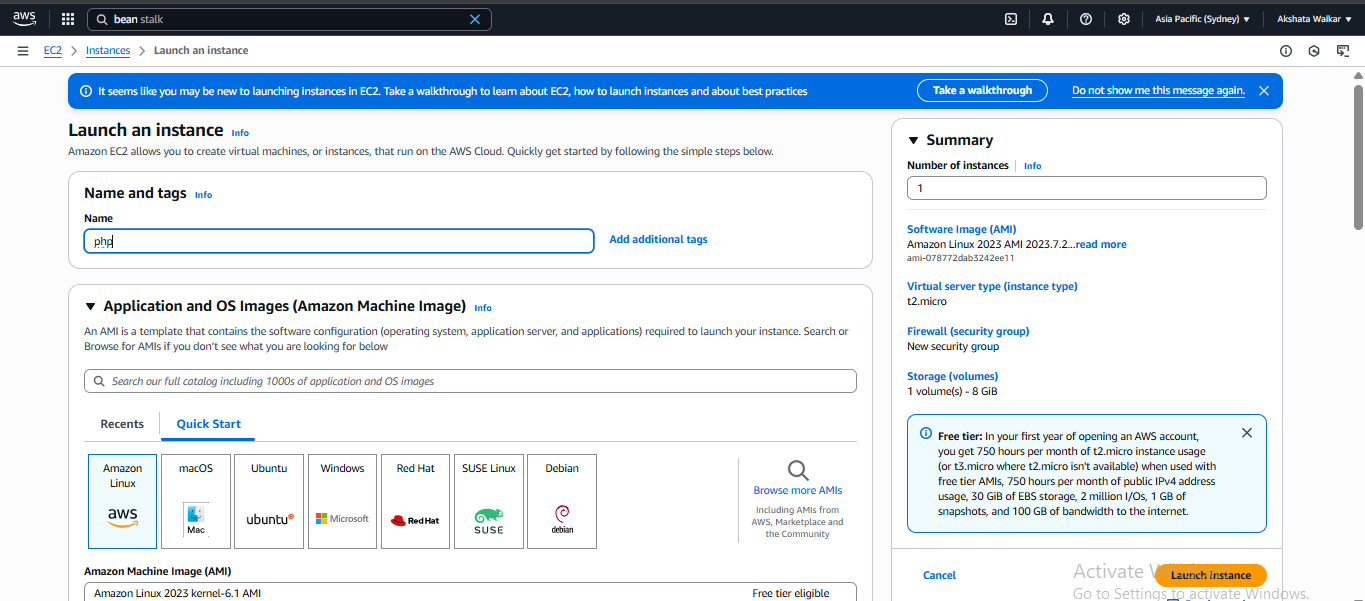
### 🔹 ****Step 4: Launch the Environment****

* Click **Create environment**
* Wait for ~5–10 minutes until the environment is fully provisioned



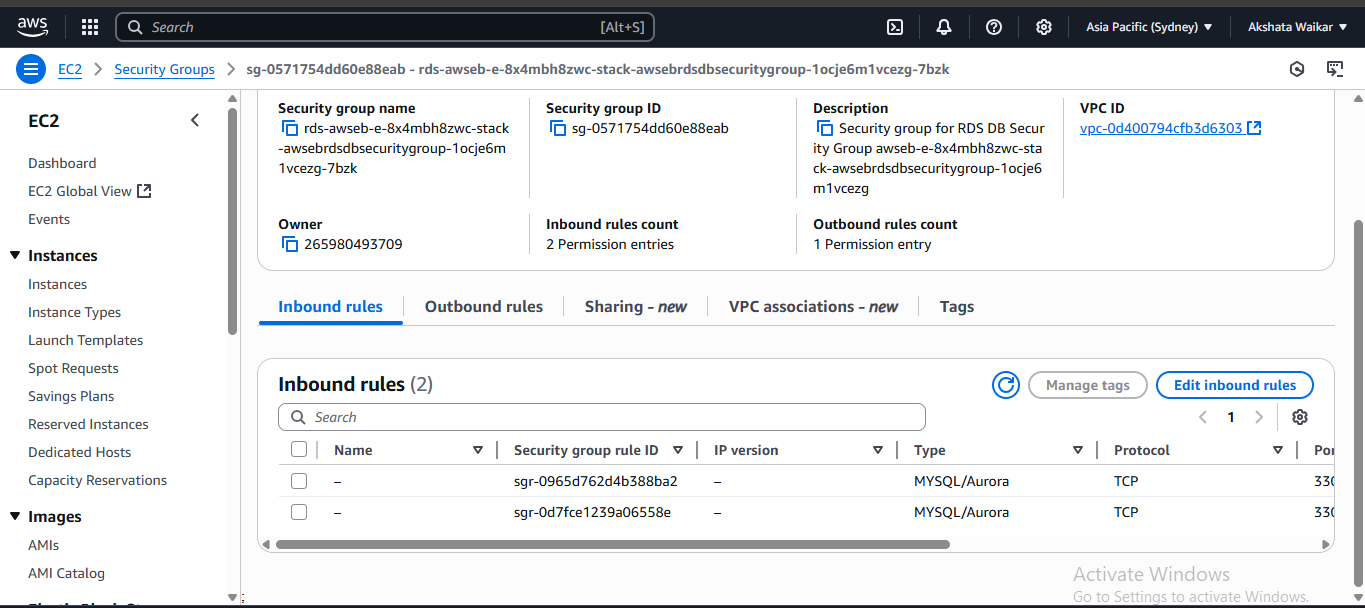
### 🔹 ****Step 5: Launch EC2 Instance in Same VPC****

1. Go to: [EC2 Console](https://console.aws.amazon.com/ec2)
2. Click **Launch Instance**
3. Fill details:
   * **Name**: rds-client-ec2
   * **AMI**: Choose **Amazon Linux 2**
   * **Instance type**: t2.micro (Free Tier eligible)
   * **Key pair**: Choose the same key pair used in Beanstalk
   * **Network**: Select **same VPC** as Elastic Beanstalk
   * **Subnet**: Select a **public subnet**
   * **Auto-assign Public IP**: Enable ✅



### 🔹 ****Step 6: Modify RDS Security Group****

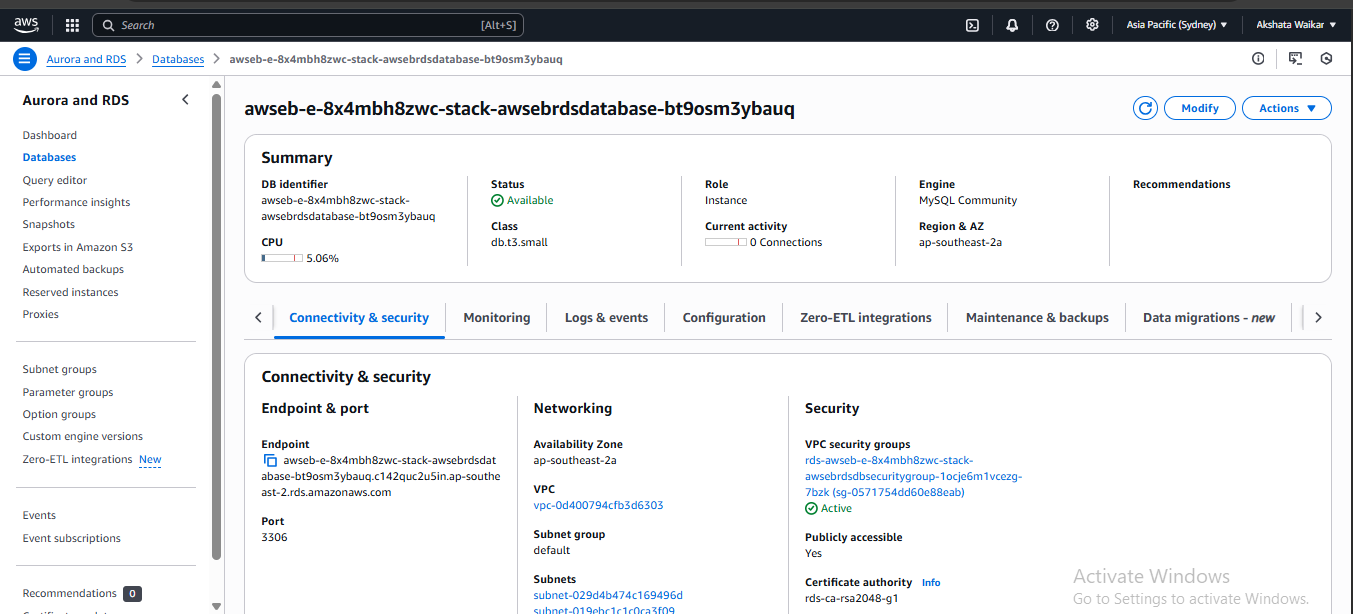
1. Go to **EC2 → Security Groups**
2. Find the **RDS security group** (used by Beanstalk)
3. Add **inbound rule**:
   * Type: MySQL/Aurora
   * Port: 3306
   * Source: EC2 security group (from Step 5)



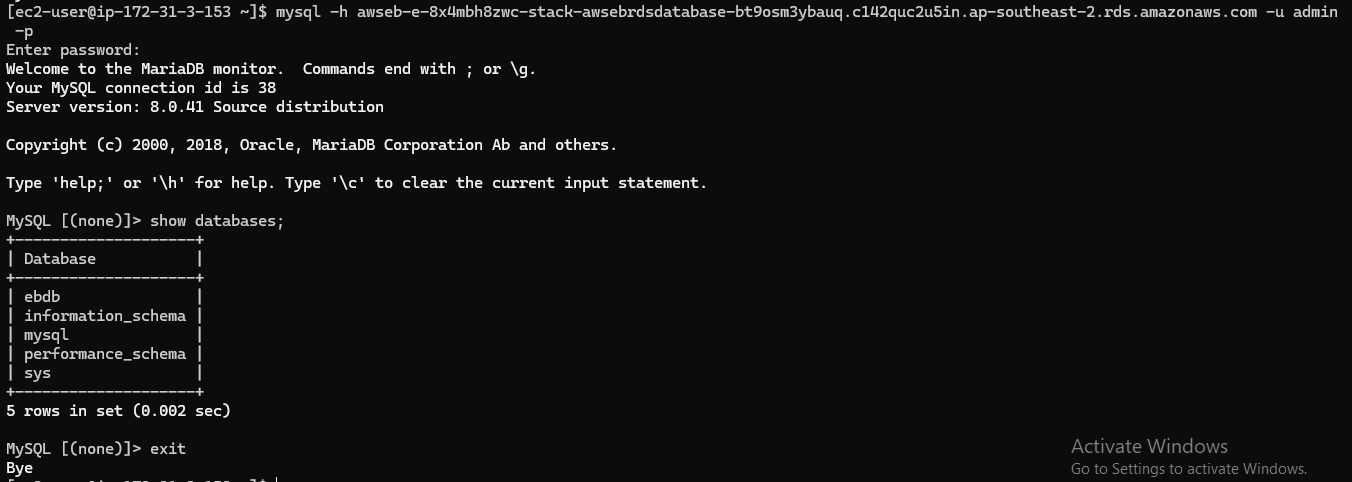
🔹 **Step 7: Connect to EC2 via SSH**

🔹 **Step 8: Install MySQL Client**

🔹 **Step 9: Get RDS Endpoint**

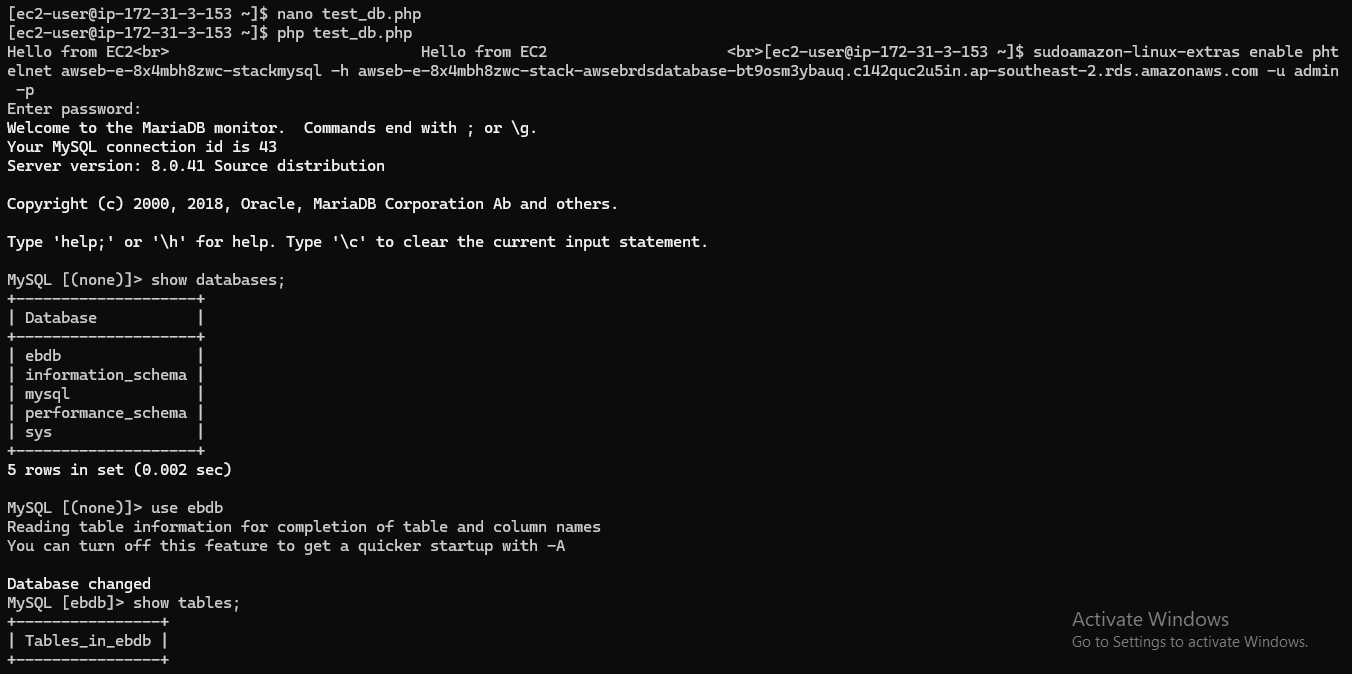


🔹 **Step 10: Connect to RDS**



🔹 **Step 11: Create test\_db.php script**

🔹 **Step 12: Run test\_db.php script**



**Conclusion :**

This project successfully demonstrates how to deploy a scalable and secure web application infrastructure using AWS-managed services. By using **Elastic Beanstalk**, we simplified the deployment and management of a PHP-based web application. Integrating **Amazon RDS (MySQL)** during environment creation provided a reliable and scalable backend database.

Additionally, by setting up an **Amazon EC2 instance** in the same VPC and securely configuring **security groups**, we enabled safe and restricted access to the RDS instance for administrative and testing purposes. This approach mirrors real-world best practices for application architecture, security, and DevOps workflows.

Through this project, we learned how to:

* Provision and configure cloud-native infrastructure
* Enable secure service-to-service communication in AWS
* Manage credentials, networking, and access controls
* Use EC2 for backend operations and testing
* Build the foundation for a 3-tier application architecture on AWS

**GitHub Repository link** : https://github.com/Akshata-Waikar/AWS-Beanstalk

**Prepared By:**  
**Akshata Waikar**  
Cloud & DevOps Intern  
Cravita Technologies