WordPress Deployment on EC2 with Amazon RDS (MySQL)

Abhishek Bhagat

 Abhishek Bhagat

💡 Introducing Today's Project!

What the project is about?

This project showcases how I deployed a **WordPress application** on an **EC2 instance**, using **Amazon RDS** for the backend MySQL database. It follows a multi-tier architecture approach - separating the application and database layers - which improves scalability, maintainability, and security.

🛠️ How I Used AWS Services in This Project

- **Amazon EC2**: Hosted Apache, PHP, and WordPress on Amazon Linux.
- **Amazon RDS (MySQL)**: Used as the backend database for persistent storage.
- **Security Groups**: Controlled and restricted access between EC2 and RDS.
- **MySQL CLI**: Used on EC2 to connect to RDS and manually create the WordPress database and user.
- **Manual Setup**: WordPress downloaded and configured manually.

😮 One Thing I Didn't Expect...

Initially, WordPress failed to connect to the database. After some debugging, I realized that the **RDS security group needed an inbound rule** allowing MySQL (port 3306) traffic **only from the EC2 security group** not just any IP. Once I corrected the rule and updated wp-config.php, the connection worked perfectly.

⏱️ This Project Took Me...

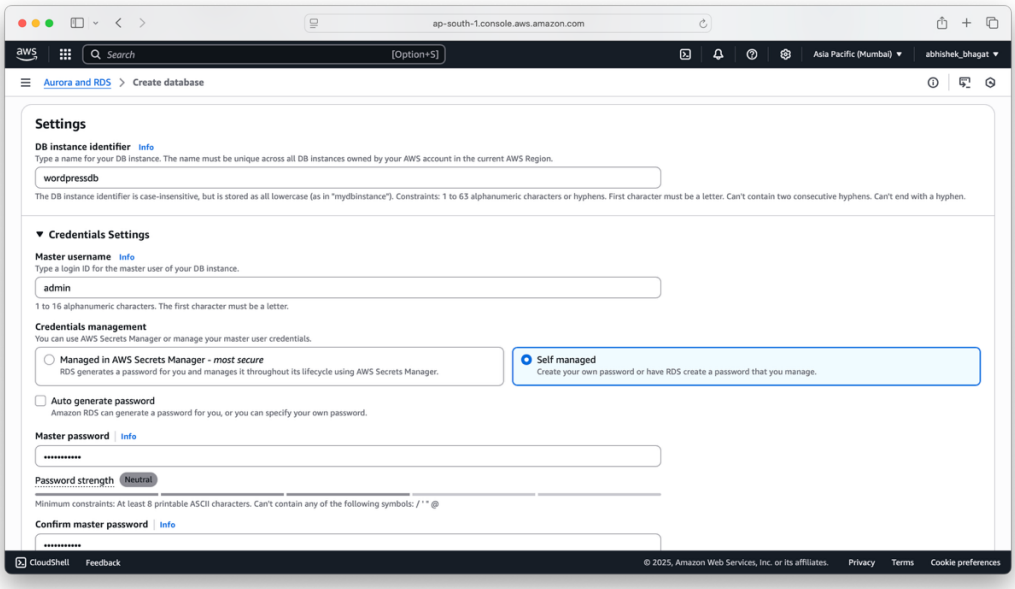
~45 minutes to provision the infrastructure, install packages, troubleshoot connection errors, and fully deploy WordPress.

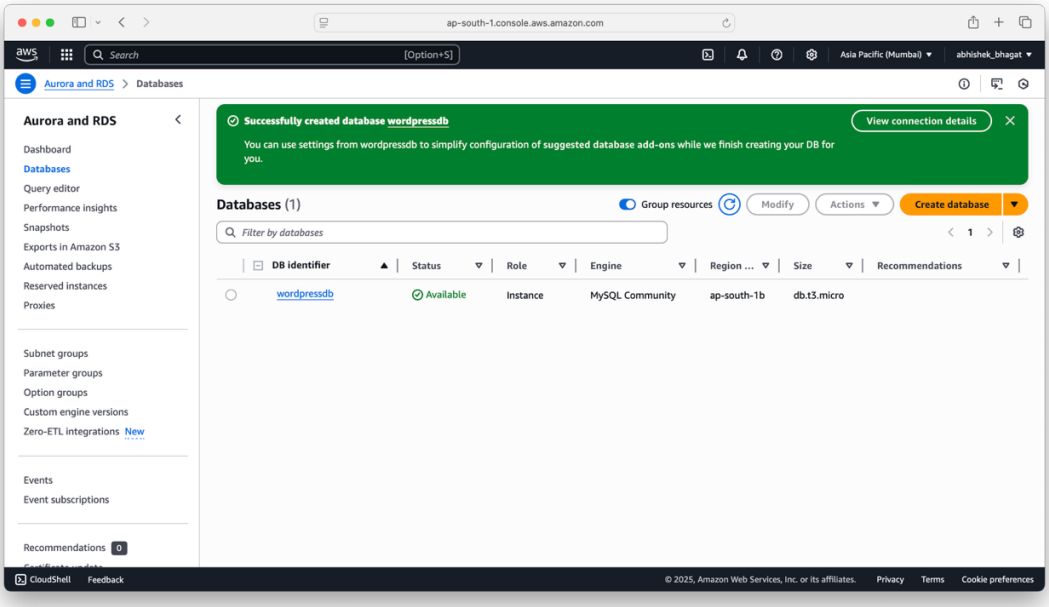
Abhishek Bhagat

📦 Project Steps Breakdown

1. Created RDS Instance (MySQL)

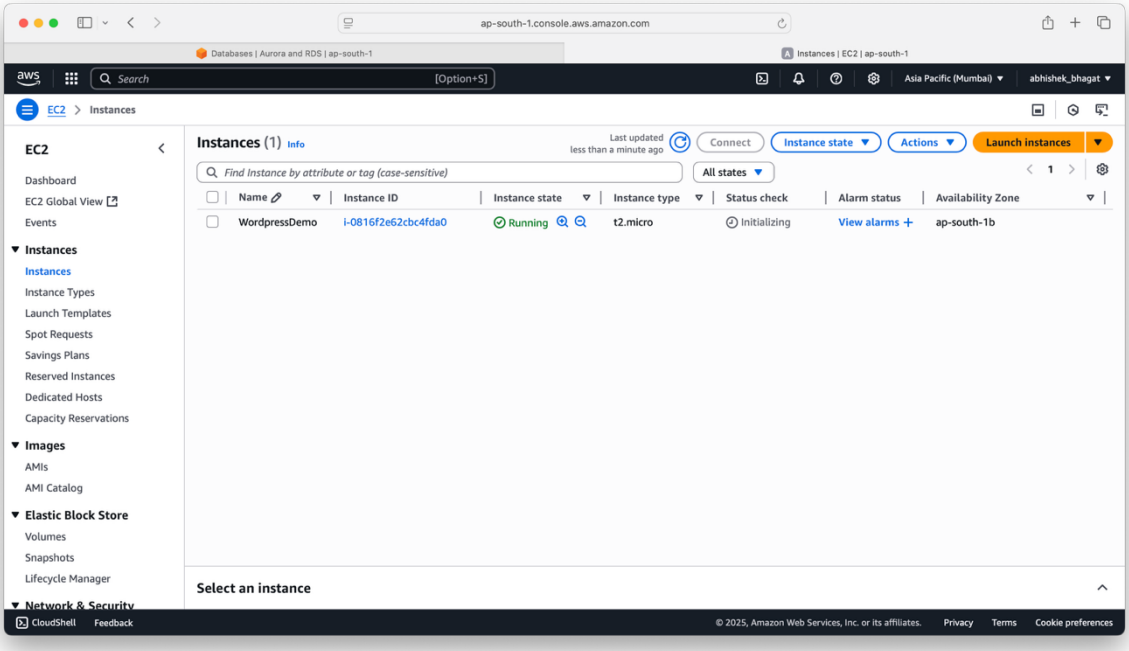
- MySQL engine
- Username: wpuser, Password: yourpassword
- DB Name: wordpressdb
- Public access: **Disabled**
- Security group: Inbound port 3306 from EC2 SG only

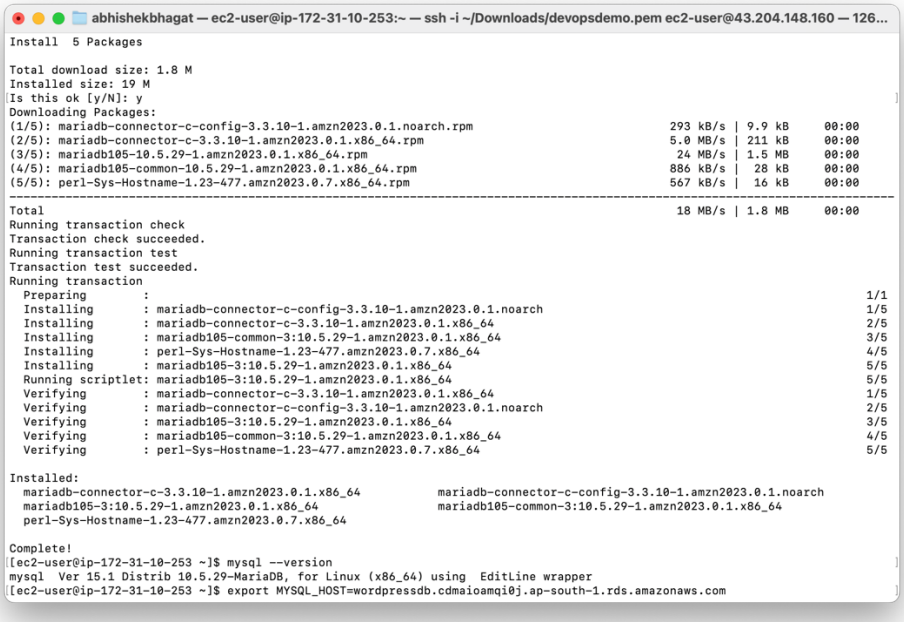




2. Launched EC2 Instance

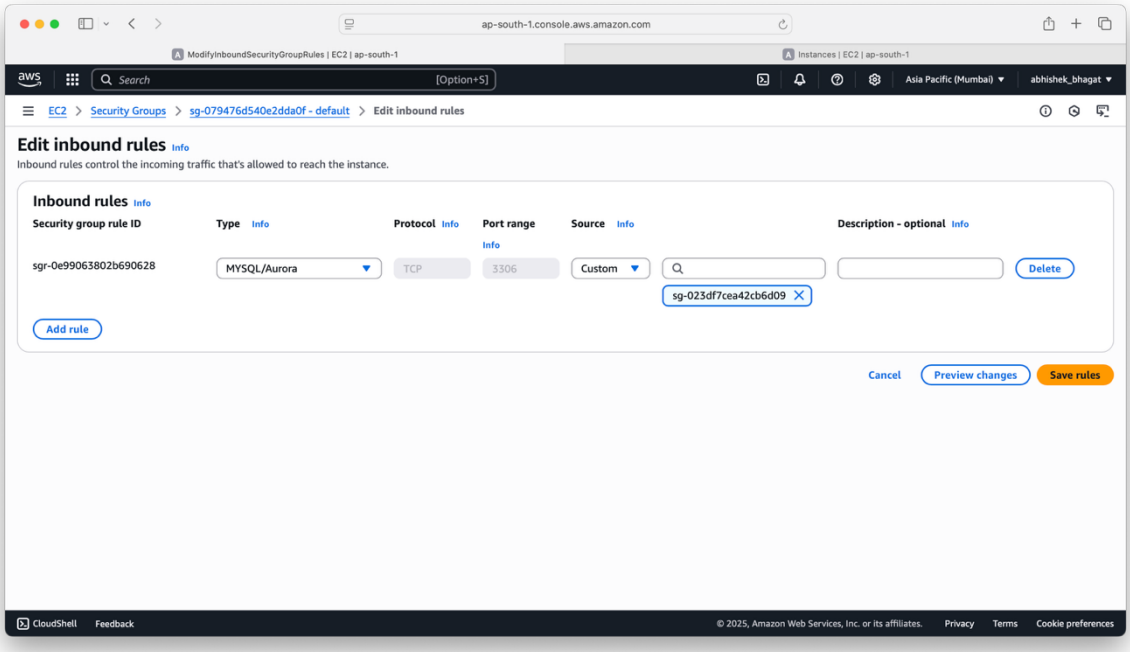
- Amazon Linux 2023 AMI
- Connected via SSH using downloaded .pem key
- Installed MySQL client for DB setup





3. Connected RDS Instance to EC2

- Allow your EC2 instance to access your Amazon RDS Instance by changing the inbound rules of the Security Group.



4. Connect to RDS and Create WordPress User

- Find your RDS endpoint under Connectivity & security.
- Set an environment variable:

```
export MYSQL_HOST=<your-rds-endpoint>
```

- Connect to MySQL with your master credentials:

```
mysql --host=$MYSQL_HOST --user=<master_user> \
  --password=<master_password> \
  --database=wordpress
```

- In the MySQL console, run:

```
CREATE USER 'wordpress' IDENTIFIED BY '<strong_password>';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress';
FLUSH PRIVILEGES;
```

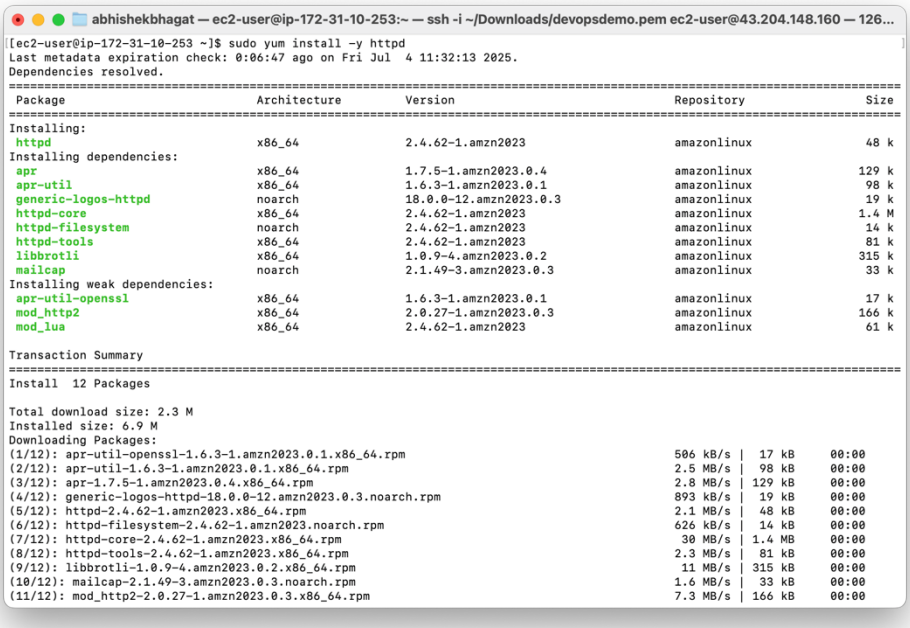
5. Installed Apache, PHP

```
sudo yum update -y

sudo yum install -y httpd php php-fpm php-json php-mysqlnd

sudo systemctl enable httpd

sudo systemctl start httpd
```



6. Installed and Configured WordPress

```
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
sudo cp -r wordpress/* /var/www/html/
sudo chown -R apache:apache /var/www/html/
cd /var/www/html
sudo cp wp-config-sample.php wp-config.php
sudo nano wp-config.php
```

🛠️ In wp-config.php, I updated:

```
define('DB_NAME', 'wordpressdb');
define('DB_USER', 'wpuser');
define('DB_PASSWORD', 'yourpassword');
define('DB_HOST', '<rds-endpoint>:3306');
```

Abhishek Bhagat

❗ Issues Faced

- WordPress page failed to load the first time due to DB connection issues.
- Fixed by adjusting RDS security group and verifying WordPress config file.
- Also had to ensure file permissions were correct in /var/www/html.
- Restarted Apache Server `sudo systemctl restart httpd`

✅ Final Outcome

- Successfully deployed a fully working WordPress site using Amazon EC2 and RDS.
- Frontend served via Apache and backend data persisted in a managed MySQL database - all with secure access control and scalable architecture.