Project in AWS
Practice Lab

Set Up a WordPress Site Using EC2 and RDS

Andra-Diana Popescu

2025

ABOUT THIS LAB

Amazon Relational Database Service (Amazon RDS) allows users to easily create, operate, and scale a relational database in the cloud. In this lab, we create an RDS database, install a web server and configure WordPress to connect to the RDS database. We then run the final configuration through the web browser and are presented with a working WordPress blog. By the end of this lab, the user will understand how to create an RDS database and configure WordPress to use it to store data.

LEARNING OBJECTIVES

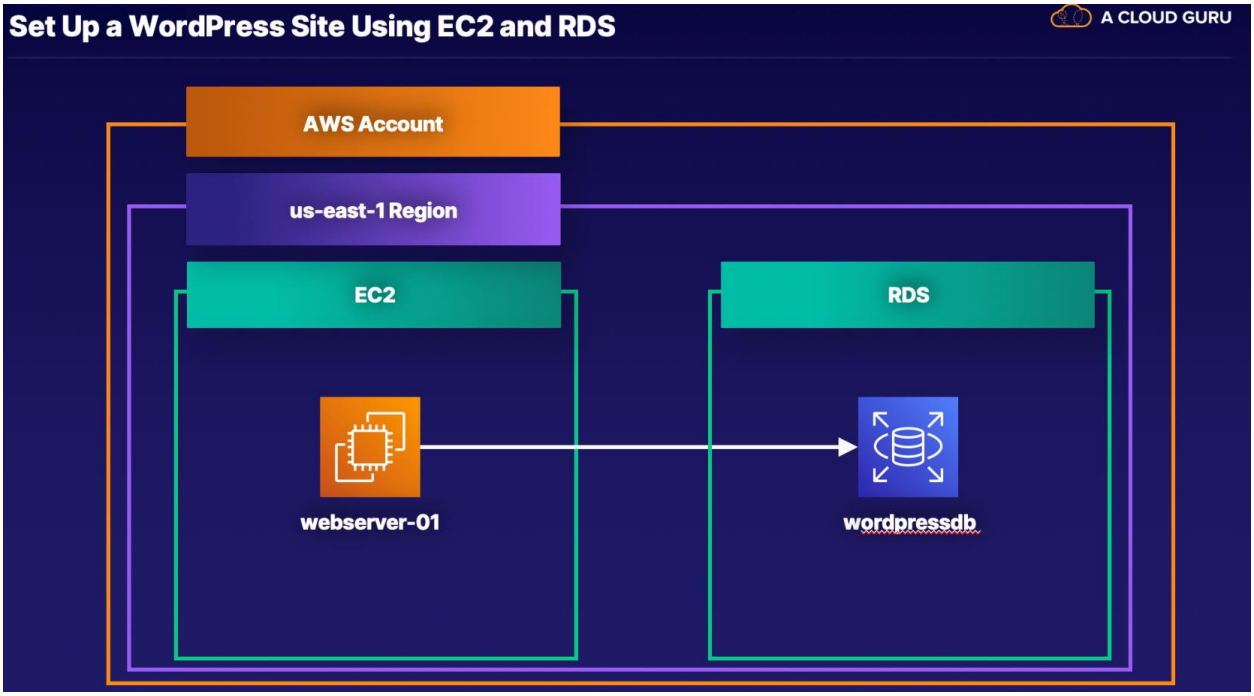
- Create an EFS File System
- Create RDS Database
- Install Apache and Dependencies
- Configure WordPress
- Modify Security Groups
- Complete WordPress Installation and Test

AWS Documentation about RDS: https://aws.amazon.com/rds/faqs/#Read_Replicas

Source: https://learn.acloud.guru/course/certified-solutions-architect-associate/

Table of Contents

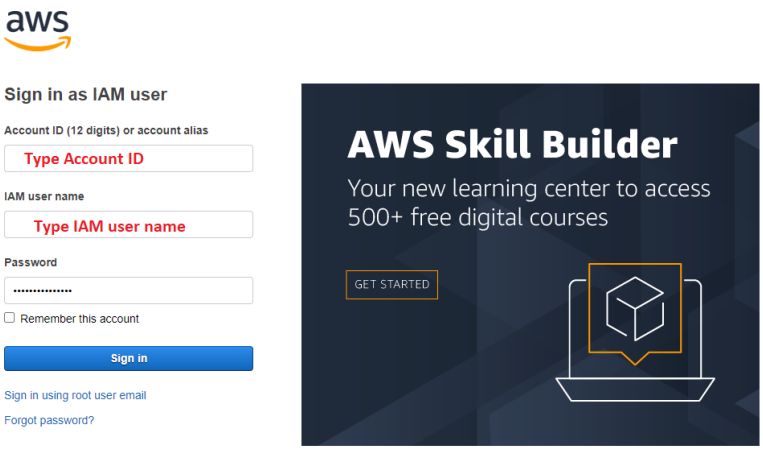
Lab Diagrams



We have the AWS account in **us-east-1** Region, and we have an EC2. Our scenario is that we are consulting for a local news outlet. They are starting their journey into online media and have decided to start with a basic blog site, and they have chosen the free WordPress blogging software package for this. They expect their blog to grow, so you've decided to use RDS to host the database for the blog's content, which can be scaled up in the future.

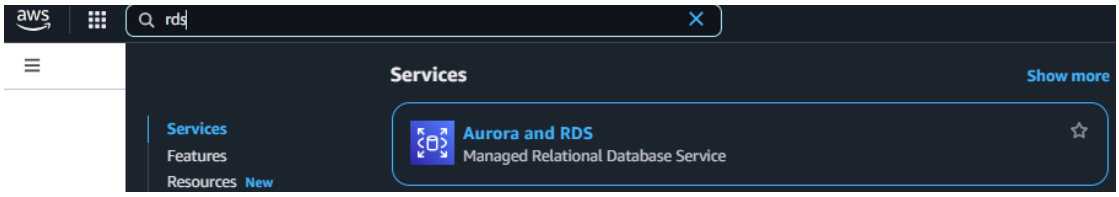
In this lab, we're going to create an RDS database, and we'll install a web server on our EC2 instance. Then, we'll install WordPress, and we'll configure it to use our RDS database to store our blog data.

Log in to your AWS account

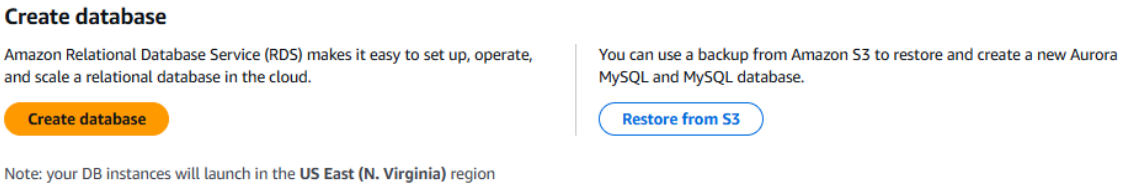


1. Create RDS Database

1. Once you are logged in to the AWS Management Console, navigate to **RDS**.

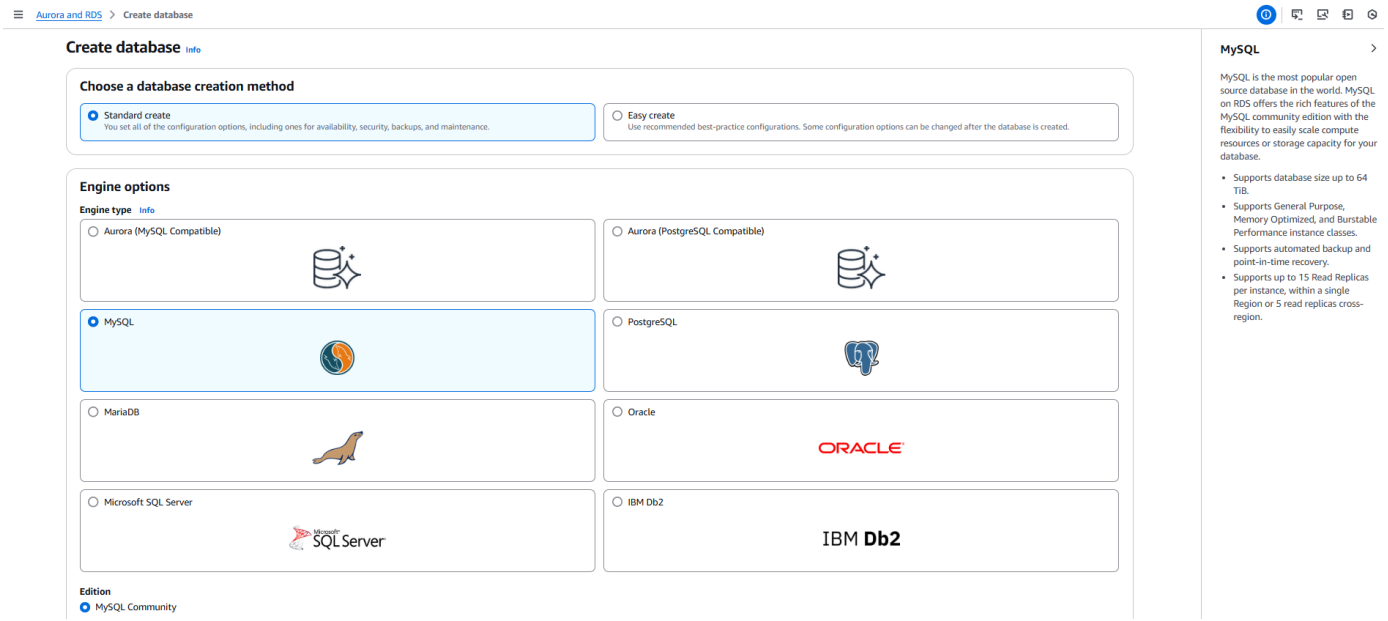


2. Click **Create database**.

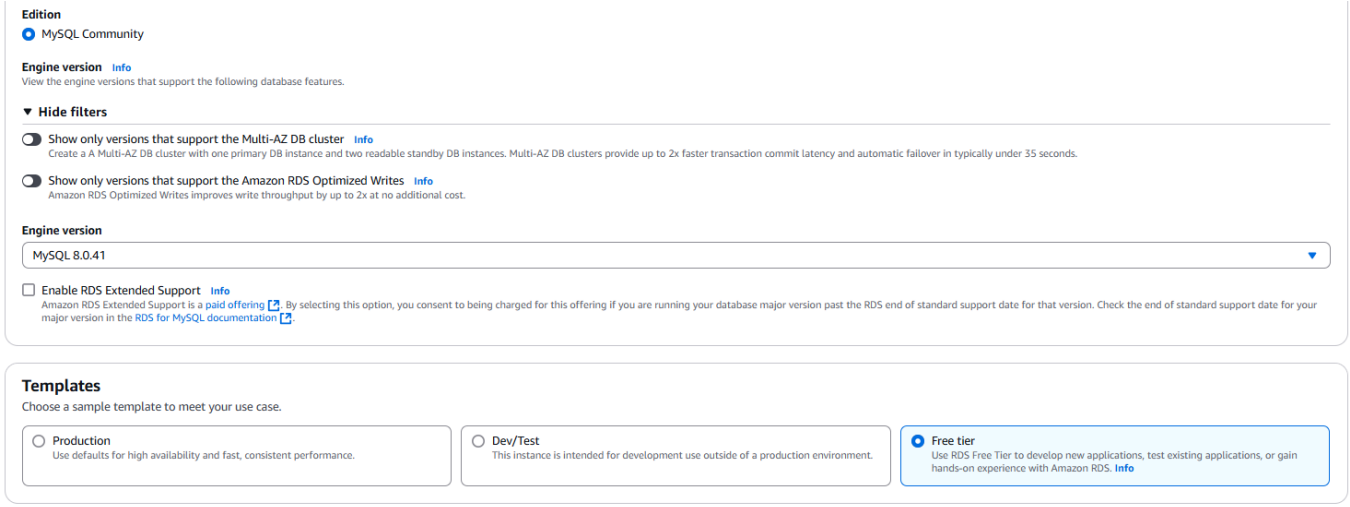


3. On the ***Create database page***, set the following parameters:

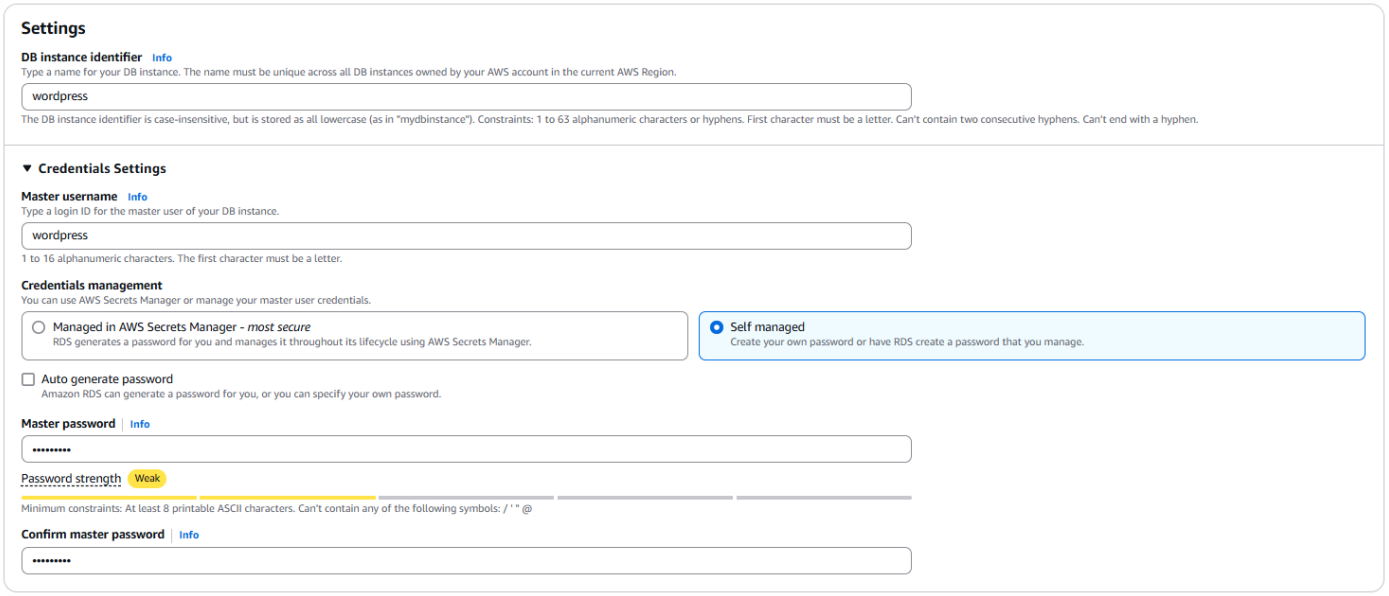
    a. Select **Standard** create.
    b. Under *Engine options*, select **MySQL**.



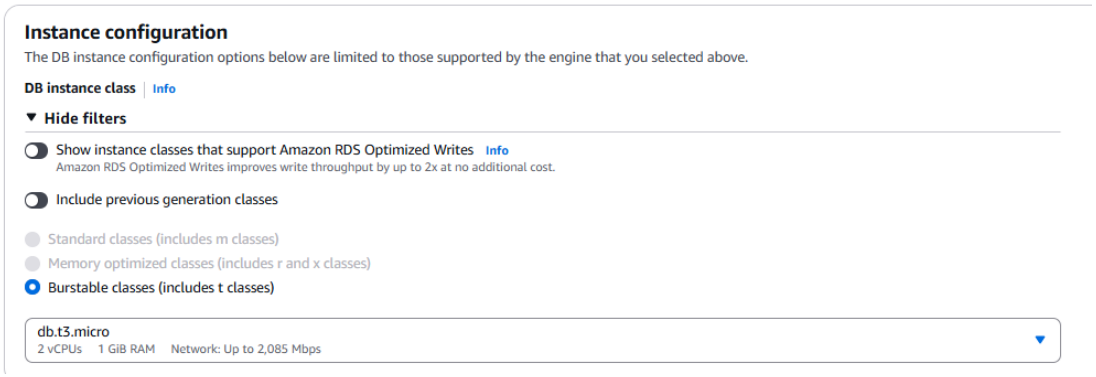
c. Under *Templates*, select **Free Tier**.



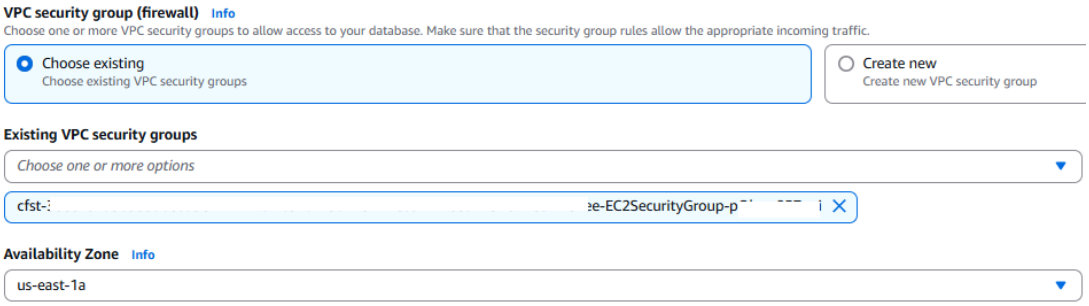
d. Under *DB instance identifier*, enter "wordpress" and copy this into your clipboard.
e. Paste in "wordpress" as the **Master username**.
f. Under *Credentials management*, select **Self managed**.
g. Paste in "wordpress" in the **Master password** and **Confirm master password** fields.



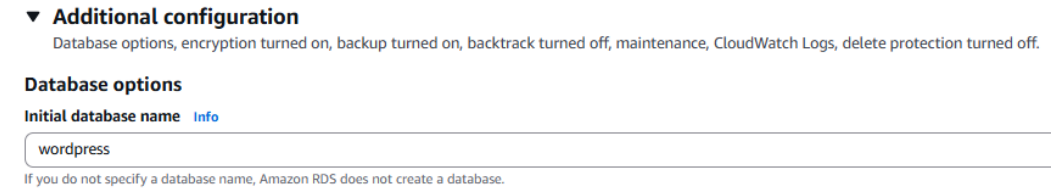
h. Under *Instance configuration*, select **Burstable classes** and make sure the class is ***db.t3.micro****.



i. Under *VPC security group*, ensure **Choose existing**.
j. Under *Existing VPC security group*, select the **non-default security group** from the dropdown menu and remove the default security group.
k. Under *Availability zone*, select **us-east-1a**.



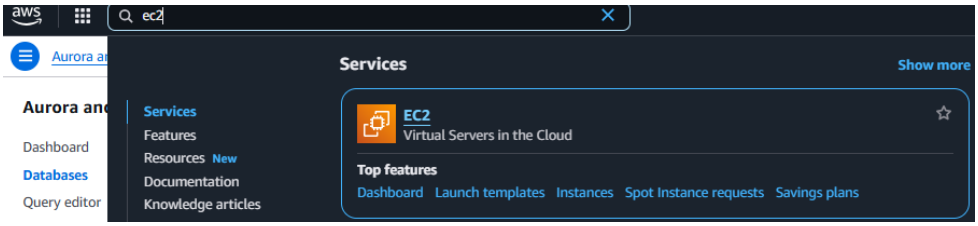
l. Expand *Additional configuration* and, under *Initial database name*, enter "wordpress".



m. Click **Create database**.

4. While the database is created, enter "ec2" in the search bar on top.

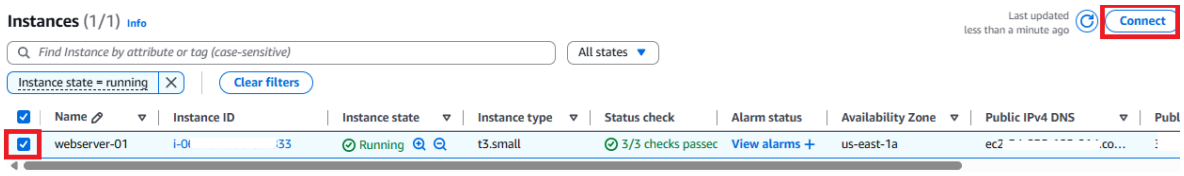
5. From the results, right-click **EC2** and open it in a new browser window or tab.



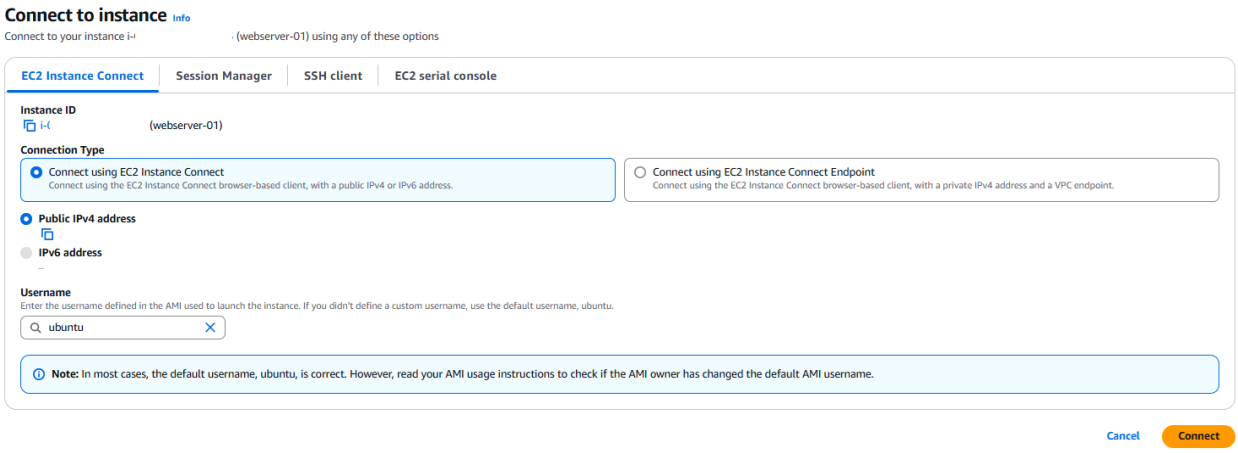
6. Under ***Resources***, click **Instances (running)**.

7. Click the checkbox next to ***webserver-01***.

8. In the top right, click **Connect**.



9. Click **Connect**.



2. Install Apache and Dependencies

1. In the terminal, install the Apache 2 web server, libraries, PHP, and PHP MySQL: **sudo apt install apache2 libapache2-mod-php php-mysql**
2. When prompted, press **Y** for yes and press **Enter**.

```
ubuntu@ip-          :~$ sudo apt install apache2 libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0 php7.4-cli php7.4-json php7.4-mysql
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser php-pear openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0 php-mysql
  php7.4-opcache php7.4-readline ssl-cert
0 upgraded, 19 newly installed, 0 to remove and 82 not upgraded.
Need to get 5026 kB of archives.
After this operation, 18.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

3. Go into the newly created **/var/www** directory: **cd /var/www/**
4. View the contents of the directory: **ls**
5. Put ***wordpress*** into its own folder in the **/var/www** directory that we're currently in: **sudo mv /wordpress .**
6. View the contents of the directory: **ls**
7. Move into the wordpress directory: **cd wordpress**
8. View the contents of the directory: **ls**
9. Move the Apache configuration file into **/etc/apache2/sites-enabled/** to enable the WordPress website to work from **/var/www/wordpress**: **sudo mv 000-default.conf /etc/apache2/sites-enabled/**
10. Restart the Apache 2 configuration: **sudo apache2ctl restart**

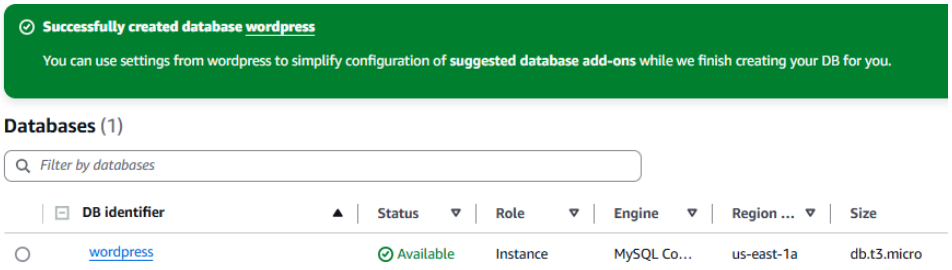
```
ubuntu@ip-[redacted]:~$ cd /var/www/
ubuntu@ip-[redacted]:/var/www$ ls
html
ubuntu@ip-[redacted]:/var/www$ sudo mv /wordpress .
ubuntu@ip-[redacted]:/var/www$ ls
html  wordpress
ubuntu@ip-[redacted]:/var/www$ cd wordpress
ubuntu@ip-[redacted]:/var/www/wordpress$ ls
000-default.conf  license.txt  wp-activate.php  wp-blog-header.php    wp-config-sample.php  wp-content   wp-includes       wp-load.php   wp-mail.php      wp-signup.php     xmlrpc.php
index.php         readme.html  wp-admin         wp-comments-post.php  wp-config.php         wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
ubuntu@ip-[redacted]:/var/www/wordpress$ sudo mv 000-default.conf /etc/apache2/sites-enabled/
ubuntu@ip-[redacted]:/var/www/wordpress$ sudo apache2ctl restart
ubuntu@ip-[redacted]:/var/www/wordpress$
```

3. Configure WordPress

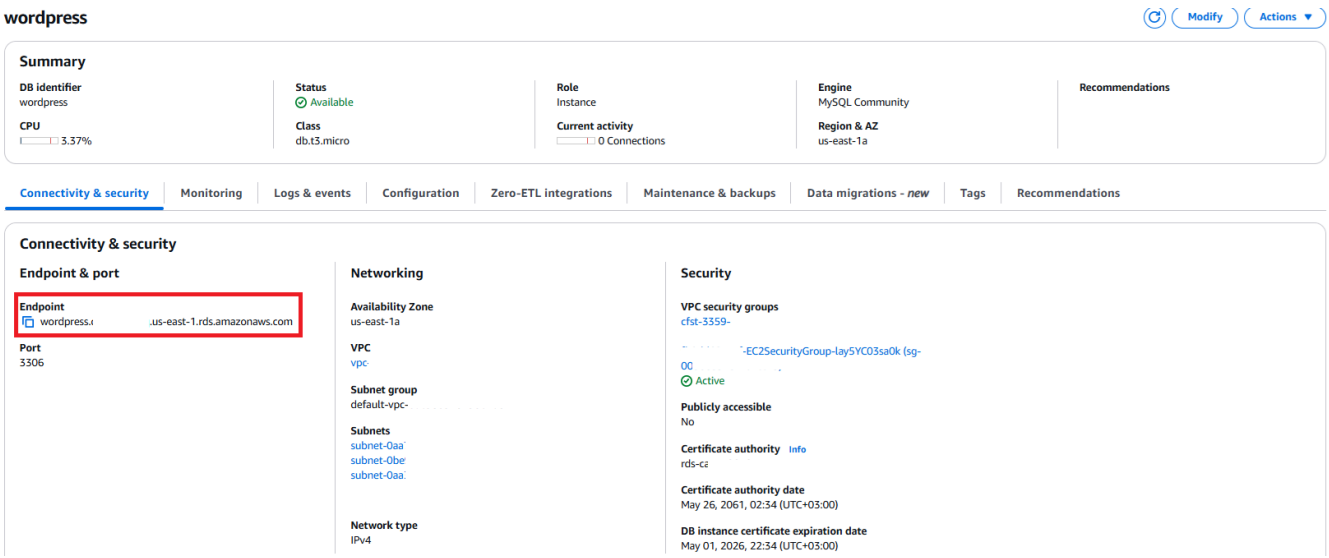
1. Open the WordPress config PHP file for editing: **sudo vi wp-config.php**
2. There is one thing here that needs to be changes, which is the MySQL hostname (that's currently set to ***localhost***, and that's incorrect). We need the endpoint from RDS.

```php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' );

/** MySQL database password */
define( 'DB_PASSWORD', 'wordpress' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

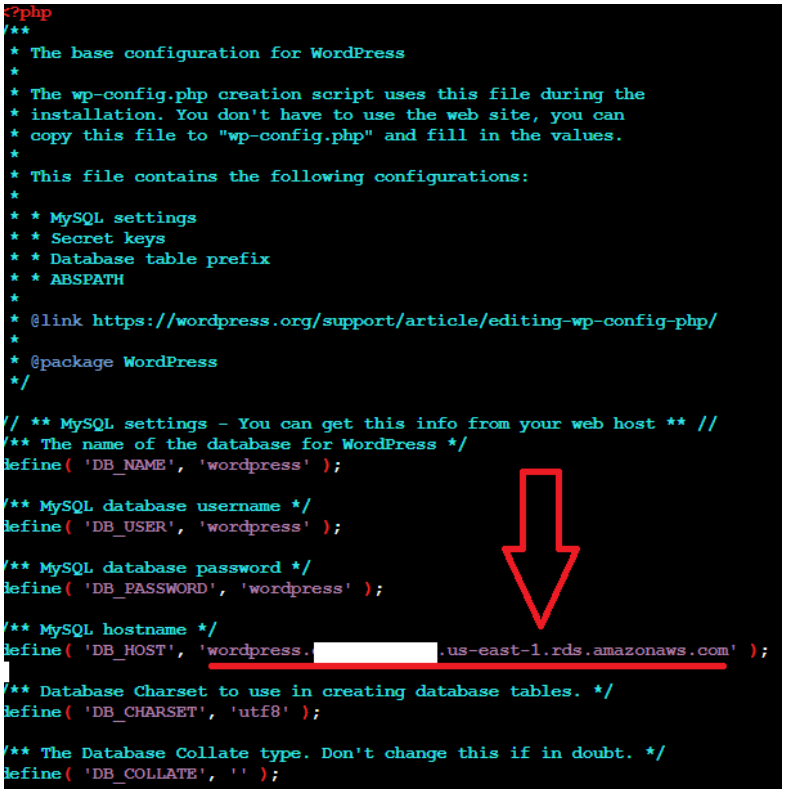
3. Return to the browser window or tab that has the RDS Databases open.
4. Click the **wordpress** database.



5. In the *Connectivity & security* tab, under *Endpoint*, copy the endpoint provided into your clipboard.

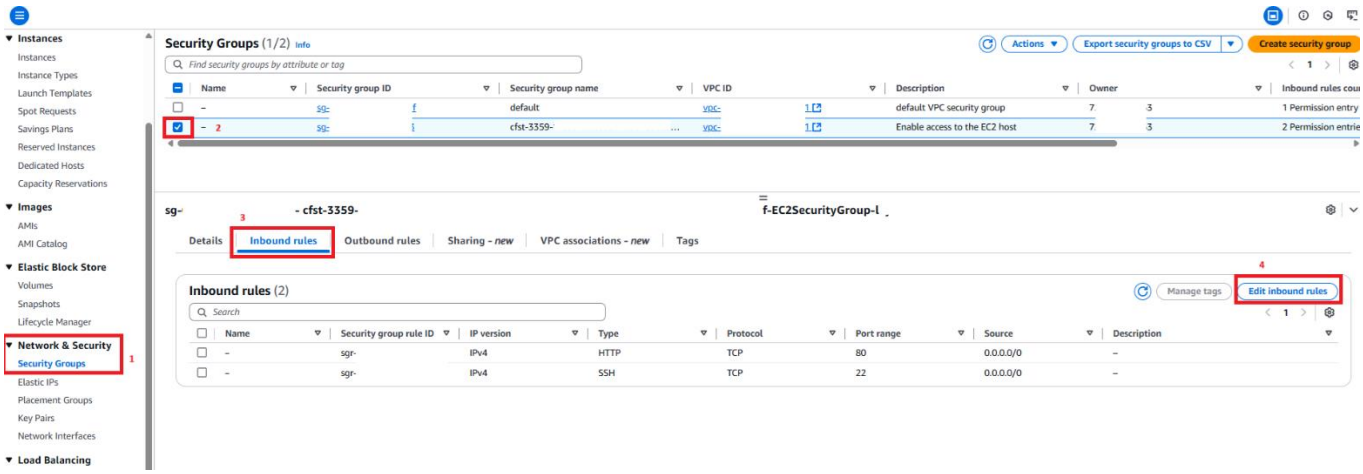


6. Return to your terminal.
7. Press “i” for INSERT.
8. Change the line **define('DB_HOST', 'localhost');** to read: **define('DB_HOST', '<INSERT ENDPOINT HERE>');**
9. Press “**esc**” and “**:wq**” to save your changes.

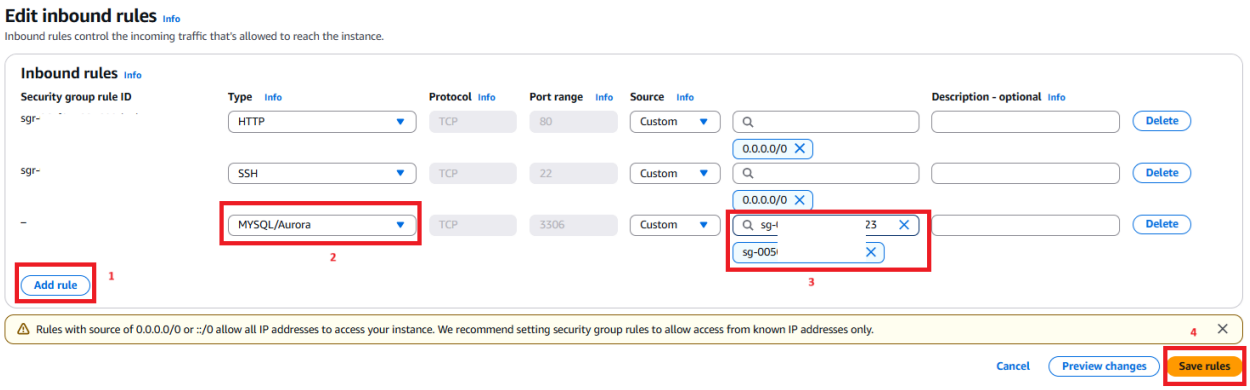


4. Modify Security Groups

1. Return to your browser window or tab with the ***EC2 Connect to instance*** page open.
2. In the left-hand navigation menu, under ***Networks & Security***, click **Security Groups**.
3. Click the checkmark next to the non-default security group among those provided in the lab.
4. Click the **Inbound rules** tab.
5. Click the **Edit inbound rules** button.

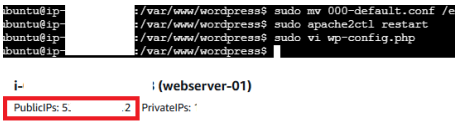


6. Click the **Add rule** button.
7. For the new rule, from the ***Type*** dropdown menu, select **MYSQL/Aurora**.
8. In the dropdown menu to the right of the **Source** column for the new rule, find and select the non-default security group. This actually means any service this security group is attached to is allowed to access MySQL. And because this security group is attached to both the EC2 instance and the MySQL database in RDS, that means they can talk to each other without any further configuration required.
9. Click **Save** rules.

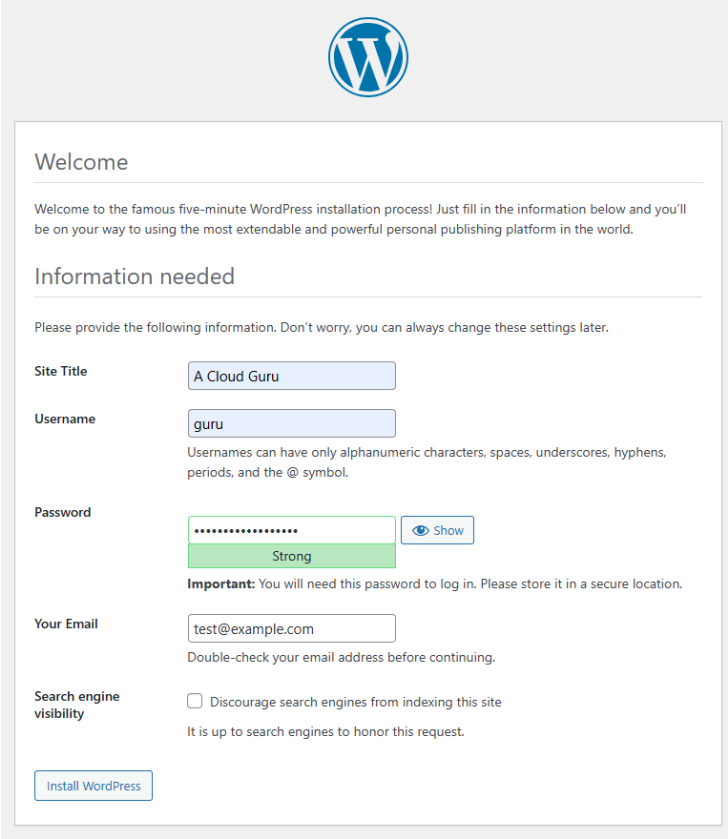


5. Complete WordPress Installation and Test

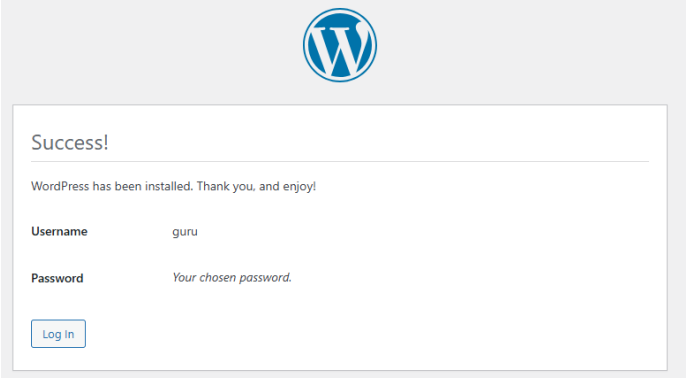
1. Return to the terminal.
2. At the bottom of the screen on the white bar, right-click the public IP now being shown after ***PublicIPs***.



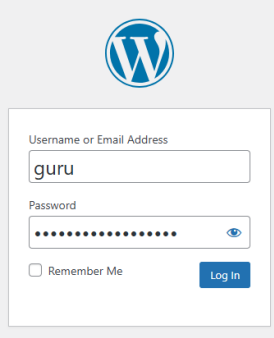
3. Click **Go** to followed by the IP address, or copy the IP address, open a new browser window or tab, and paste it there.
4. On the WordPress installation page, enter in the following information for each field:
    a. ***Site Title***: "A Cloud Guru"
    b. ***Username***: "guru"
    c. ***Password***: Select a strong password to use here, and make sure to copy it in your clipboard for later.
    d. ***Your Email***: "test@example.com"
5. Click **Install WordPress**.



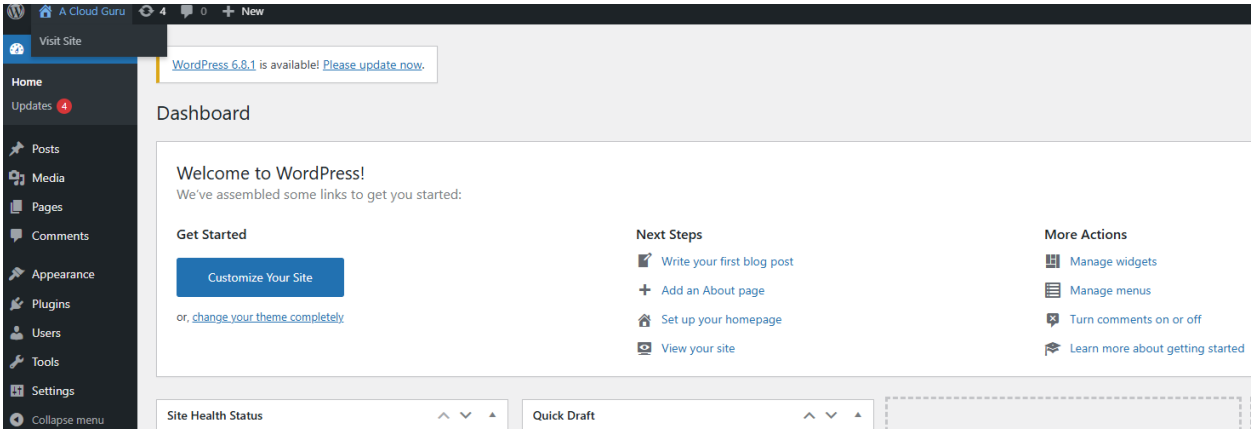
6. Click **Log in**.



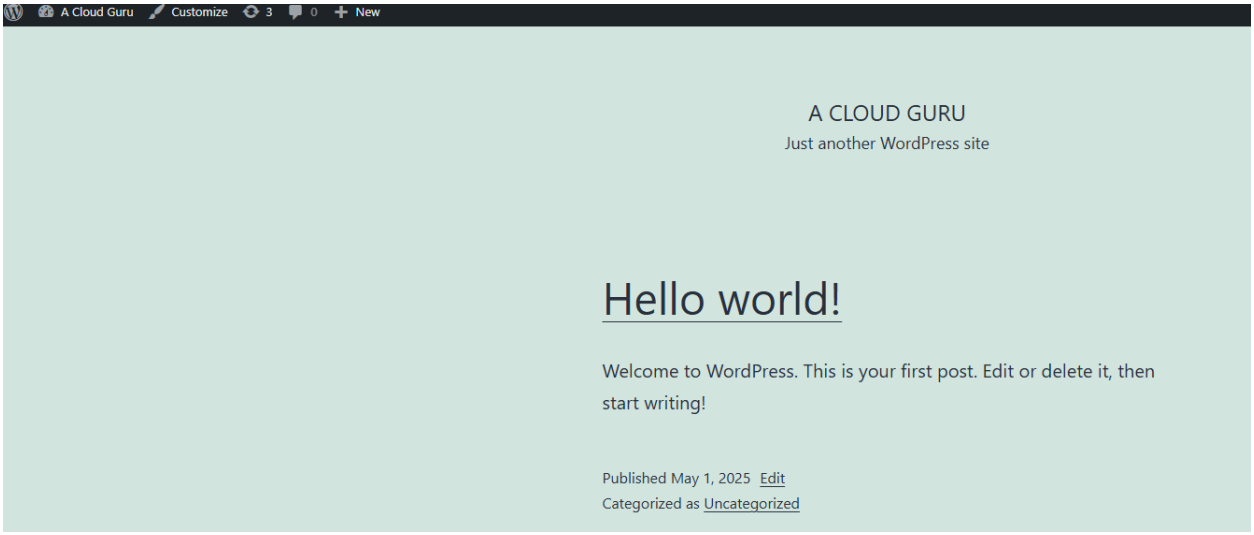
7. Enter "guru" for the *Username* or *email* and paste in the password that you copied earlier, to log in.



8. To view the website you just created, click **A Cloud Guru** in the top left corner of the page.
9. Click **Visit Site** to visit your newly created WordPress site.



10. This is where you can publish blog content.


A Cloud Guru
Customize
3
0
New
A CLOUD GURU
Just another WordPress site
Hello world!
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!
Published May 1, 2025 Edit
Categorized as Uncategorized