

**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

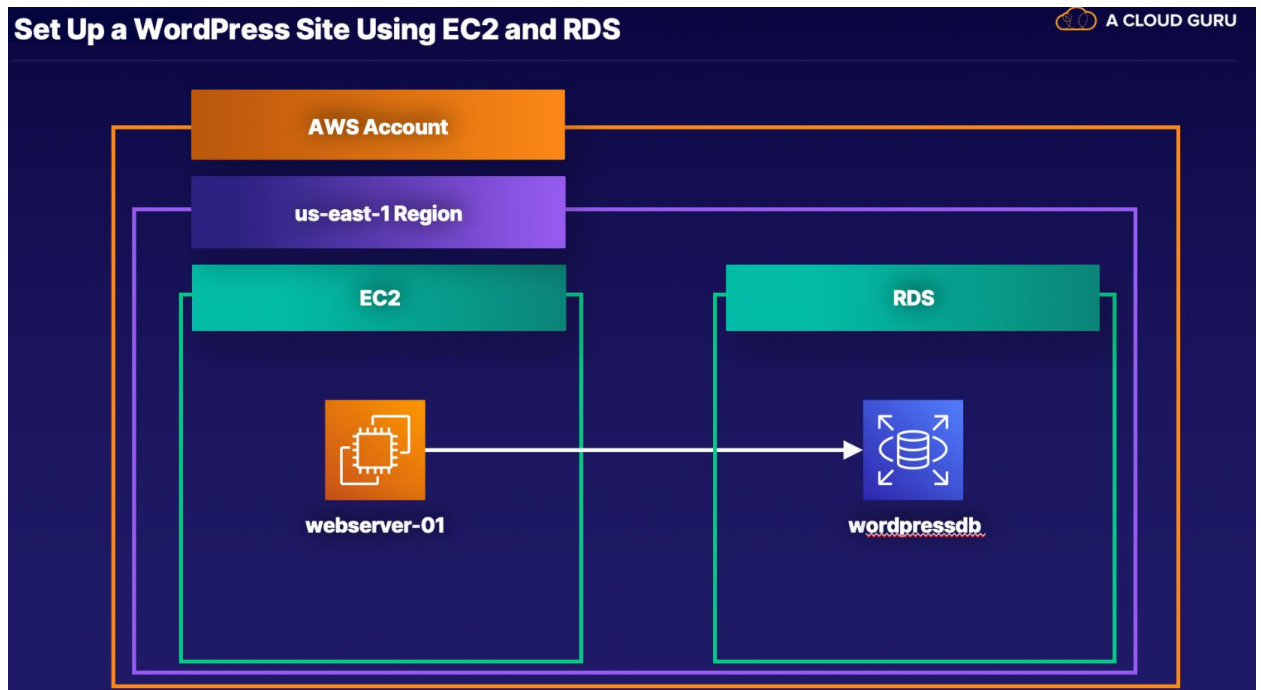
AWS Documentation about RDS: https://aws.amazon.com/rds/faqs/#Read_Replicas

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

Table of Contents

Lab Diagrams.....	4
Log in to your AWS account	5
1. Create RDS Database.....	5
2. Install Apache and Dependencies	9
3. Configure WordPress.....	10
4. Modify Security Groups	12
5. Complete WordPress Installation and Test.....	13

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



Sign in as IAM user

Account ID (12 digits) or account alias

Type Account ID

IAM user name

Type IAM user name

Password

☐ Remember this account

Sign in

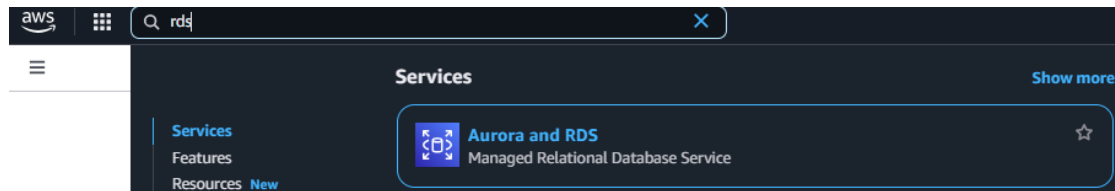
Sign in using root user email

[Forgot password?](#)



1. Create RDS Database

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



2. Click **Create database**.

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Create database

You can use a backup from Amazon S3 to restore and create a new Aurora MySQL and MySQL database.

Restore from S3

Note: your DB instances will launch in the **US East (N. Virginia)** region

3. On the **Create database page**, set the following parameters:
 - a. Select **Standard** create.
 - b. Under **Engine options**, select **MySQL**.

☰ [Aurora and RDS](#) > Create database

Create database [Info](#)


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) 


☒ **MySQL** 


☐ MariaDB 

☐ Microsoft SQL Server 

☐ Aurora (PostgreSQL Compatible) 

☐ PostgreSQL 

☐ Oracle 

☐ IBM Db2 

Edition

☒ **MySQL Community**

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

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

Edition

☒ **MySQL Community**

Engine version [Info](#)
View the engine versions that support the following database features.

▼ **Hide filters**

☐ **Show only versions that support the Multi-AZ DB cluster** [Info](#)
Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

☐ **Show only versions that support the Amazon RDS Optimized Writes** [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine version

MySQL 8.0.41 ▼

☐ **Enable RDS Extended Support** [Info](#)
Amazon RDS Extended Support is a [paid offering](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#).

Templates

Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**
This instance is intended for development use outside of a production environment.

☒ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

- 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.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

wordpress

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 Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

wordpress

1 to 16 alphanumeric characters. The first character must be a letter.

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](#)

Password strength [Info](#) **Weak**

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / * @

Confirm master password [Info](#)

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

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

▼ **Hide filters**

☒ **Show instance classes that support Amazon RDS Optimized Writes** [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

☐ **Include previous generation classes**

☐ Standard classes (includes m classes)

☐ Memory optimized classes (includes r and x classes)

☒ **Burstable classes (includes t classes)**

db.t3.micro
2 vCPUs 1 GiB RAM Network: Up to 2,085 Mbps

- 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**.

VPC security group (firewall) [Info](#)
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

☒ **Choose existing**
Choose existing VPC security groups

☐ **Create new**
Create new VPC security group

Existing VPC security groups

Choose one or more options

cfst-: de-EC2SecurityGroup-p --- i X

Availability Zone [Info](#)

us-east-1a

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

▼ Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

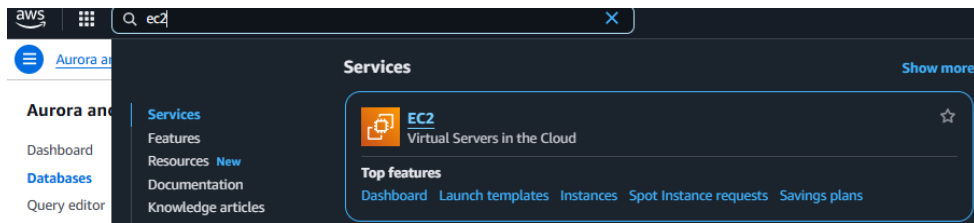
Database options

Initial database name [Info](#)

wordpress

If you do not specify a database name, Amazon RDS does not create a database.

- 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.



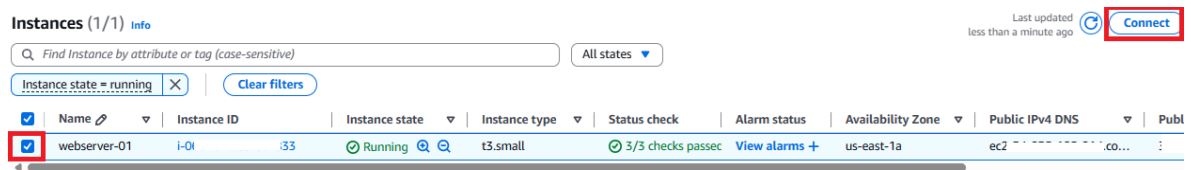
6. Under **Resources**, click **Instances (running)**.

Resources

You are using the following Amazon EC2 resources in the United States (N. Virginia) Region:

Instances (running)	1	Auto Scaling Groups	0
Dedicated Hosts	0	Elastic IPs	0
Key pairs	0	Load balancers	0
Security groups	2	Snapshots	0

7. Click the checkbox next to **webserver-01**.
8. In the top right, click **Connect**.



9. Click **Connect**.

Connect to instance [Info](#)
 Connect to your instance i- (webserver-01) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID
 i- (webserver-01)

Connection Type

☒ Connect using EC2 Instance Connect
 Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint
 Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

☒ Public IPv4 address
☐ IPv6 address

Username
 Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

Q ubuntu X

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel 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-10.0.0.1:~$ 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-10-0-1-10:~$ cd /var/www/
ubuntu@ip-10-0-1-10:/var/www$ ls
html
ubuntu@ip-10-0-1-10:/var/www$ sudo mv /wordpress .
ubuntu@ip-10-0-1-10:/var/www$ ls
html  wordpress
ubuntu@ip-10-0-1-10:/var/www$ cd wordpress
ubuntu@ip-10-0-1-10:/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-10-0-1-10:/var/www/wordpress$ sudo mv 000-default.conf /etc/apache2/sites-enabled/
ubuntu@ip-10-0-1-10:/var/www/wordpress$ sudo apache2ctl restart

```

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
/**
 * 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.

🟢 **Successfully created database wordpress**

You can use settings from wordpress to simplify configuration of suggested database add-ons while we finish creating your DB for you.

Databases (1)

DB identifier	Status	Role	Engine	Region ...	Size
wordpress	🟢 Available	Instance	MySQL Co...	us-east-1a	db.t3.micro

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

wordpress Modify Actions

Summary

DB identifier wordpress	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 3.37%	Class db.t3.micro	Current activity 0 Connections	Region & AZ us-east-1a	

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Maintenance & backups | Data migrations - new | Tags | Recommendations

Connectivity & security

Endpoint & port

Endpoint
wordpress.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1a

VPC
vpc-

Subnet group
default-vpc-

Subnets
subnet-Oaa-
subnet-Obe-
subnet-Oaa-

Network type
IPv4

Security

VPC security groups
cfst-3359-

EC2SecurityGroup-lay5YC03saOk (sg-)
Active

Publicly accessible
No

Certificate authority
Info
rds-ca-

Certificate authority date
May 26, 2061, 02:34 (UTC+03:00)

DB instance certificate expiration date
May 01, 2026, 22:34 (UTC+03:00)

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.

```
?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', 'wordpress. .us-east-1.rds.amazonaws.com' );

/** 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', '' );
```

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.

The screenshot shows the AWS Management Console interface for Security Groups. In the left-hand navigation menu, the 'Network & Security' section is expanded, and 'Security Groups' is selected. The main content area displays a list of security groups. The group 'cfst-3359-' is selected. Below the list, the 'Inbound rules' tab is active, showing two existing rules: HTTP on port 80 and SSH on port 22. An 'Edit inbound rules' button is visible in the top right of the rules list.

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**.

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

The screenshot shows the 'Edit inbound rules' form in the AWS Management Console. The form displays three existing rules: HTTP, SSH, and a new rule for MySQL/Aurora. The 'Add rule' button is highlighted with a red box and a '1'. The 'MySQL/Aurora' option in the 'Type' dropdown is highlighted with a red box and a '2'. The 'sg-005' option in the 'Source' dropdown is highlighted with a red box and a '3'. At the bottom, the 'Save rules' button is highlighted with a red box and a '4'.

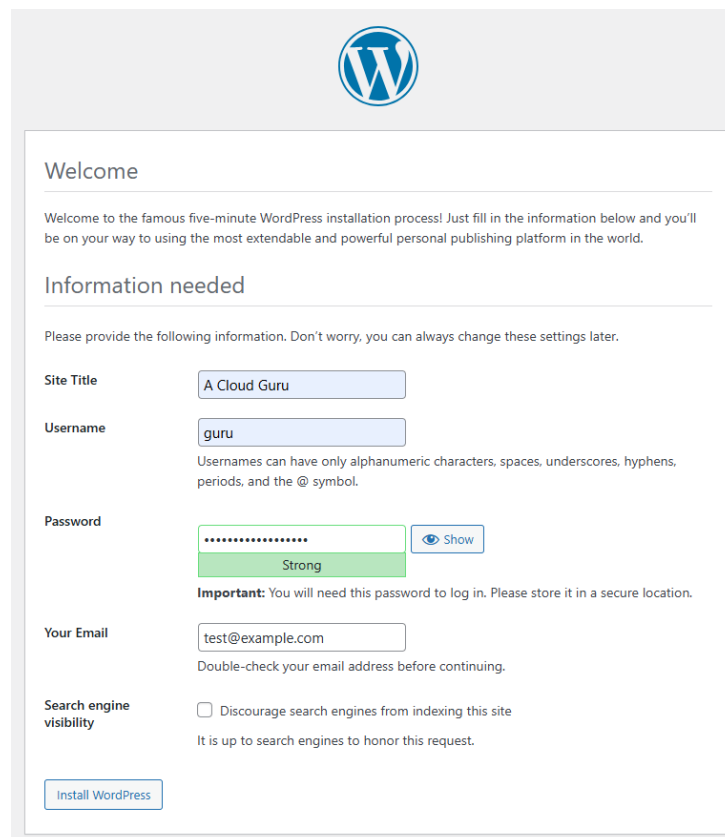
5. Complete WordPress Installation and Test

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**.

```
buntu@ip:~$ /var/www/wordpress$ sudo mv 000-default.conf /e
buntu@ip:~$ /var/www/wordpress$ sudo apache2ctl restart
buntu@ip:~$ /var/www/wordpress$ sudo vi wp-config.php
buntu@ip:~$ /var/www/wordpress$

i- (webserver-01)
PublicIPs: 5. 2 PrivateIPs: 1
```

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**.



The image shows the WordPress installation form. At the top is the WordPress logo. Below it is a 'Welcome' section with a message about the five-minute installation process. The 'Information needed' section follows, with a note that settings can be changed later. The form fields are: 'Site Title' (A Cloud Guru), 'Username' (guru), 'Password' (a strong password with a 'Show' button), 'Your Email' (test@example.com), and 'Search engine visibility' (a checkbox to discourage indexing). At the bottom is an 'Install WordPress' button.

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title: A Cloud Guru

Username: guru

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: [password field] [Show]

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email: test@example.com

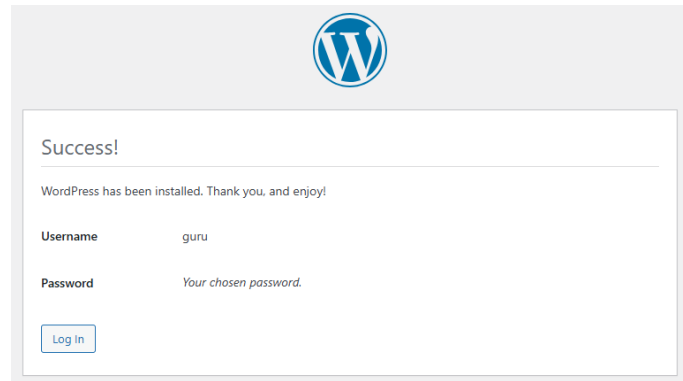
Double-check your email address before continuing.

Search engine visibility: ☐ Discourage search engines from indexing this site

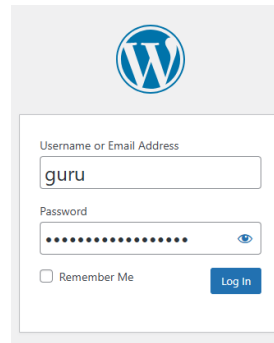
It is up to search engines to honor this request.

Install WordPress

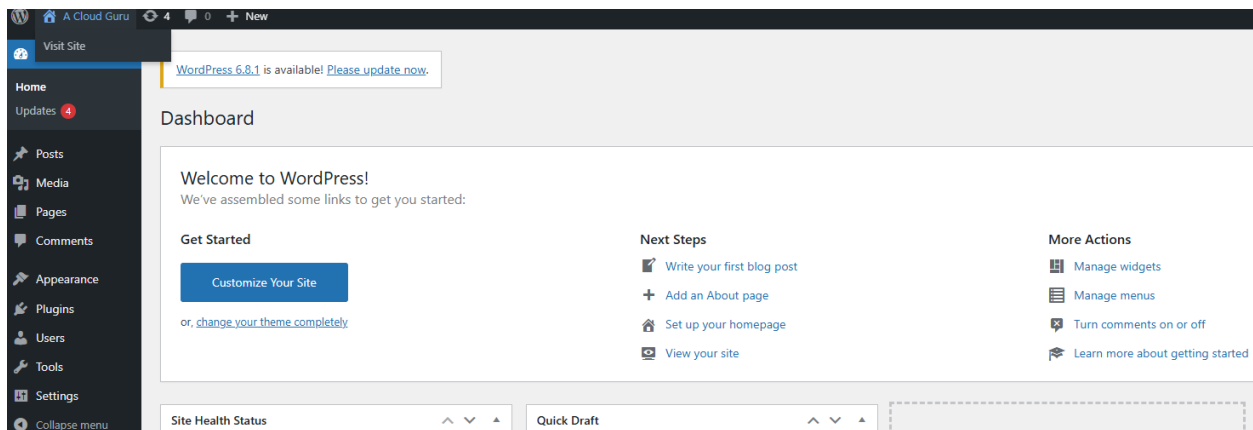
6. Click **Log in**.



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
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](#)