Deploying a 3 − Tier WordPress Application on AWS

TABLE OF CONTENTS

- Introduction to the Project
- Set up an RDS instance
- Launch an EC2 instance
- Install and configure Apache httpd
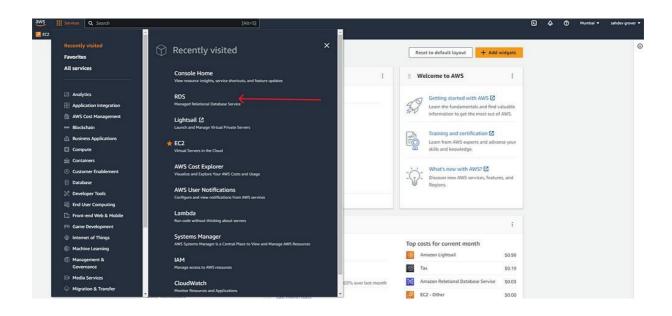- To Install and configure WordPress
- Test the website
- Conclusion

INTRODUCTION

Looking to create a website or blog but don't know where to begin? AWS Amazon Web Services) provides a variety of services and tools that can help you build and deploy your website easily and flexibly. One of the most widely used web applications that can be run on AWS is WordPress, a powerful and widely used CMS. In this project, I will guide you through the step-by-step process of deploying a 3-tier WordPress application on AWS, utilizing user-friendly services such as EC2 and RDS. Upon completion of this guide, you will have a working WordPress site hosted on AWS, and the knowledge and confidence to customize and manage it according to your preferences.
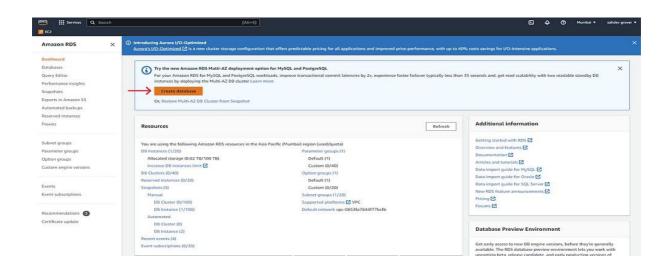
A project of launching an EC2 instance, installing a webserver of Apache, installing PHP code to run WordPress, and connecting it to a database on AWS.

To set up an RDS database for your 3- tier WordPress application, follow these steps:
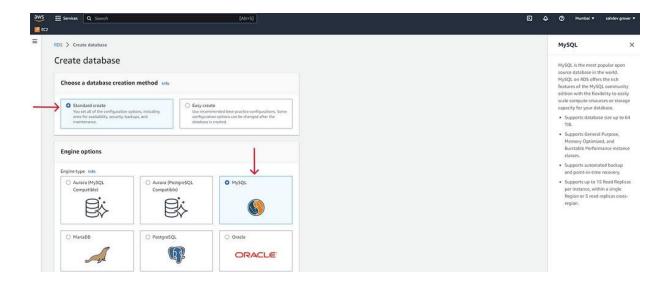
Login to the AWS Management Console and navigate to RDS.

Click on "Create database"
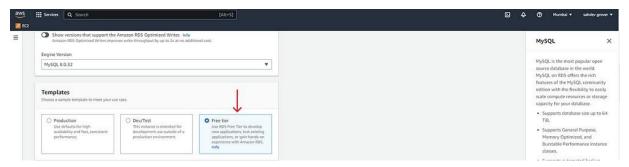
Choose the database (such as MySQL, PostgreSQL, Oracle, SQL Server, MariaDB, or Amazon Aurora).
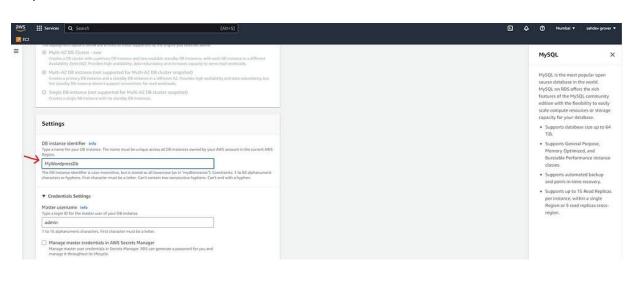
Select the version and edition of the database engine you want to use.

Engine Version

MySQL 8.0.28 ▼

Edition
◉ MySQL Community

ⓘ **Known issues/limitations**
Review the Known issues/limitations ☑ to learn about potential compatibility issues with specific database versions.

Choose the templates you want to use for the database

Configure the database settings such as the database name, username, and password

Choose the appropriate instance class based on your performance and storage needs.

Select the appropriate VPC and subnet group to launch the database in.

Connectivity Info

Compute resource
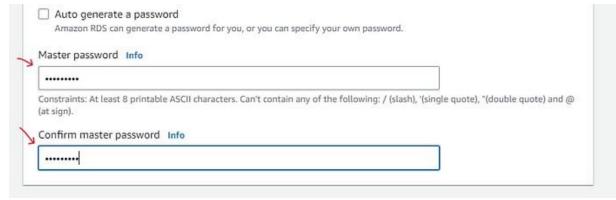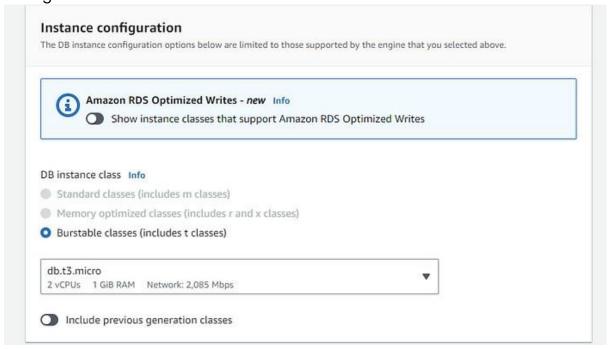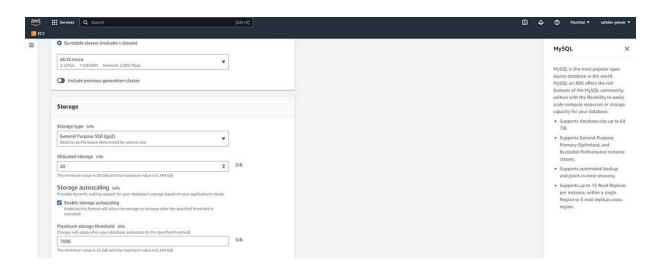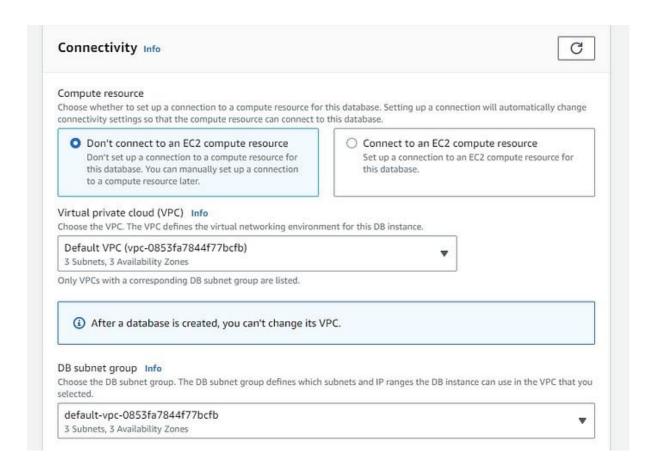
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

- ● **Don't connect to an EC2 compute resource**
  Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

- ○ **Connect to an EC2 compute resource**
  Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

| Default VPC (vpc-0853fa7844f77bcfb) |
| 3 Subnets, 3 Availability Zones |

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

DB subnet group Info

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

| default-vpc-0853fa7844f77bcfb |
| 3 Subnets, 3 Availability Zones |

Configure the database security group to control access to the database.

Public access Info

- ● **Yes**
  RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

- ○ **No**
  RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

Click on Create Database.

(i) You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel   **Create database**

Review and confrm the settings, then launch the database instance.

AFTER YOU CREATED YOUR DATABASE.

CLICK ON IT AND YOU WILL GET THE BELOW DETAILS FOR YOUR DATABASE CONNECTION.
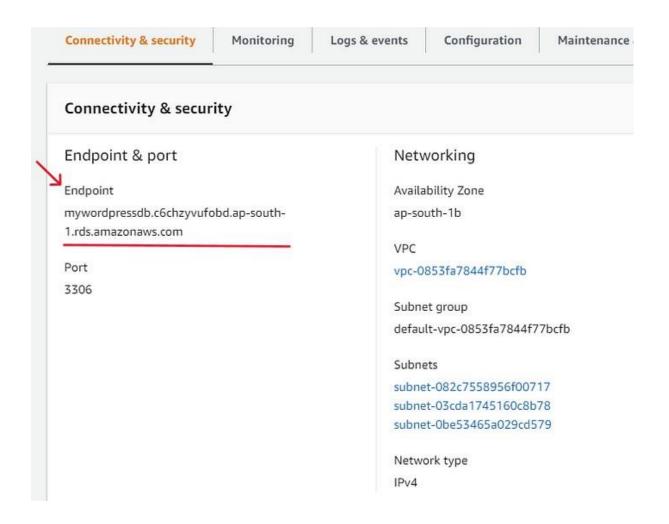
DB_NAME — your_database_name DB_USER —
your_database_username (MASTER USERNAME)

DB_PASSWORD —
your_database_password (MASTER PASS.)

DB_HOST —
your_database_endpoint_url
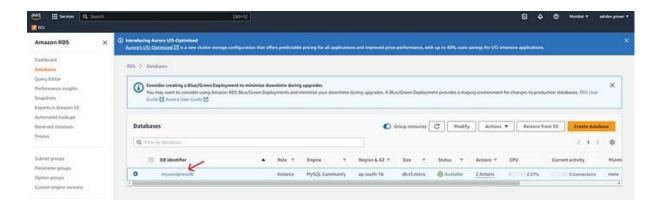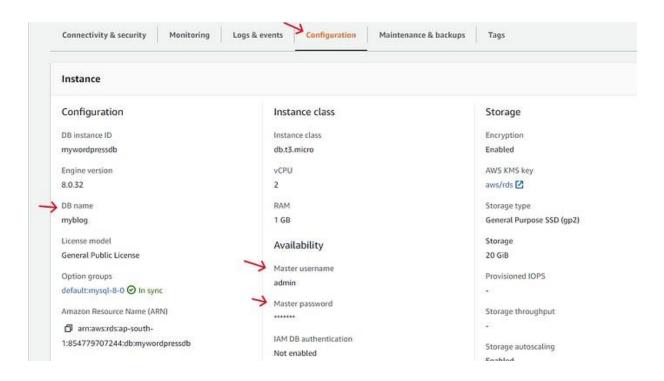
Connectivity & security

Endpoint & port

Endpoint
mywordpressdb.c6chzyvufobd.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1b

VPC
vpc-0853fa7844f77bcfb

Subnet group
default-vpc-0853fa7844f77bcfb

Subnets
subnet-082c7558956f00717
subnet-03cda1745160c8b78
subnet-0be53465a029cd579

Network type
IPv4

Deploy a WordPress website on an Amazon EC2 instance.

step 1. Launch an EC2 instance using Amazon Linux -2 AMI
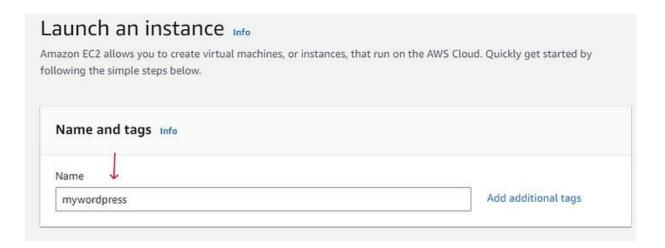
Log in to your AWS Management Console.

Click on the "EC2" service to open the EC2 Dashboard.

Click the "Launch Instance" button.
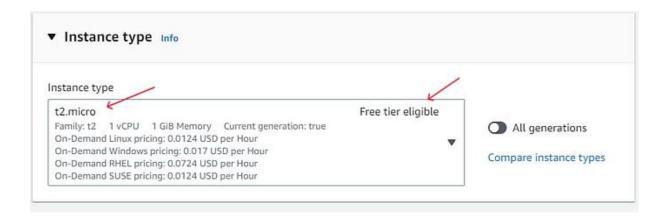
Give the name/tag to your instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

mywordpress

Add additional tags
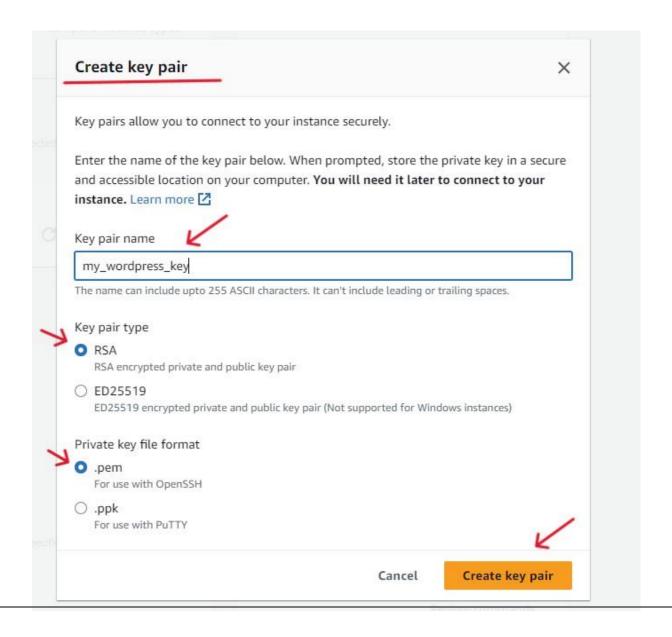
Choose an Amazon Machine Image AMI from the provided list. This is the operating system for your instance.
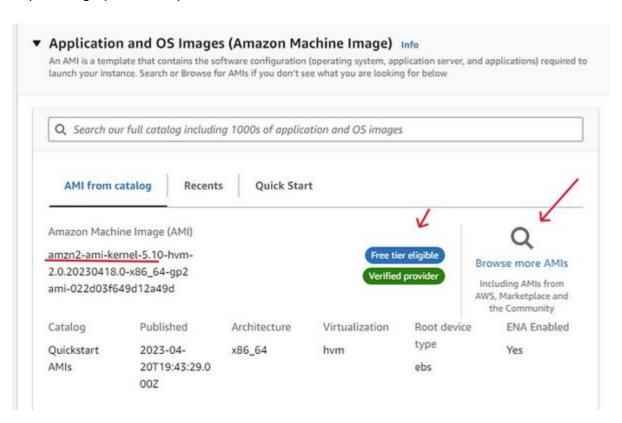
▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

AMI from catalog	Recents	Quick Start

Amazon Machine Image (AMI)

amzn2-ami-kernel-5.10-hvm-
2.0.20230418.0-x86_64-gp2
ami-022d03f649d12a49d

Free tier eligible
Verified provider

Q
Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
Quickstart AMIs	2023-04-20T19:43:29.000Z	x86_64	hvm	ebs	Yes

Select an Instance Type. An instance type determines the hardware of the host computer that AWS will allocate to your instance.
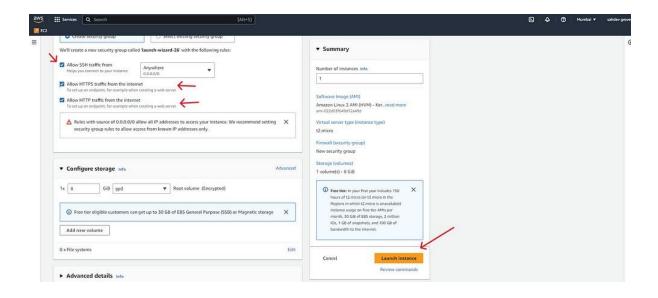
Select a key pair, and for that Create a key pair and download it to your computer. This key pair is used to authenticate your instance and give you access to it via SSH.

Configure Instance Details such as Network, subnet, security groups, and storage. Configure the Security Group settings to allow traffic to your instance. By default, a security group is created that allows SSH access from any IP address. However, you can customize this to fit your needs.

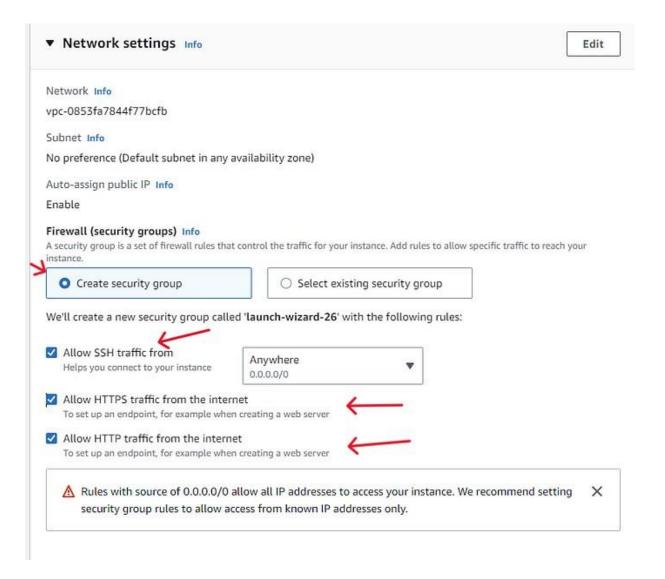▼ **Network settings** Info   Edit

Network Info
vpc-0853fa7844f77bcfb

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group   ○ Select existing security group

We'll create a new security group called '**launch-wizard-26**' with the following rules:

☑ Allow SSH traffic from   Anywhere
  Helps you connect to your instance   0.0.0.0/0   ▼

☑ Allow HTTPS traffic from the internet
  To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
  To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting   ✕
  security group rules to allow access from known IP addresses only.

Review and launch the instance. Here, you can review the settings you have chosen for your instance. If everything looks good, click on the "Launch" button.
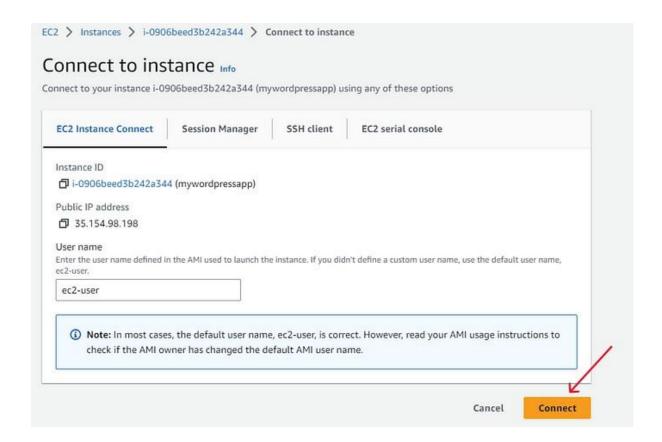
Your instance will now be launching.

It may take a few minutes for your instance to launch after your instance launch, you connect your instance
Click on the instance you want to connect and apply to connect

Connect to instance Info

Connect to your instance i-0906beed3b242a344 (mywordpressapp) using any of these options

| EC2 Instance Connect | Session Manager | SSH client | EC2 serial console |

Instance ID
  🗗 i-0906beed3b242a344 (mywordpressapp)

Public IP address
  🗗 35.154.98.198

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

```
ec2-user
```

ⓘ **Note:** In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel   **Connect**

After connecting to the instance, you will come to the terminal. NOW, Follow the below steps

Step 2: Install and start the Apache httpd web server.

yum install httpd -y

systemctl enable httpd — now

```
[root@ip-172-31-33-213 ~]# systemctl enable httpd --now
[root@ip-172-31-33-213 ~]#
```

```
[ec2-user@ip-172-31-33-213 ~]$ sudo su -
[root@ip-172-31-33-213 ~]# yum install httpd -y
Amazon Linux 2023 repository                                          34 kB/s | 3.6 kB    00:00
Amazon Linux 2023 Kernel Livepatch repository                        27 kB/s | 2.9 kB    00:00
Amazon Linux 2023 Kernel Livepatch repository                       971 kB/s | 156 kB    00:00
Package httpd-2.4.56-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-33-213 ~]#
```

Step 3: Go to the Apache web server web page location and download

WordPress and extract it.

```
[root@ip-172-33-213 ~]# cd /var/www/html
[root@ip-172-33-213 html]# ls
index.html
[root@ip-172-33-213 html]# pwd
/var/www/html
[root@ip-172-33-213 html]# wget https://wordpress.org/latest.tar.gz
--2023-05-11 15:14:48--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23018887 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz              100%[===================================>]  21.95M  6.65MB/s    in 4.8s

2023-05-11 15:14:54 (4.62 MB/s) - 'latest.tar.gz' saved [23018887/23018887]

[root@ip-172-33-213 html]#
```

write this command, you can get this one from the WordPress downloads

your files will be extracted using this command

```
tar -xzvf latest.tar.gz
```

Step 4: Copy all WordPress contents in the Apache httpd web page location.

```
[root@ip-172-31-33-213 html]# cp  -r wordpress/*  .
[root@ip-172-31-33-213 html]# pwd
/var/www/html
[root@ip-172-31-33-213 html]# ls
index.php     license.txt  wordpress      wp-admin         wp-comments-post.php  wp-content  wp-includes   wp-load.php   wp-mail.php      wp-signup.php     xmlrpc.php
latest.tar.gz readme.html  wp-activate.php wp-blog-header.php wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
[root@ip-172-31-33-213 html]#
```

Step 5: Make Apache owner for all WordPress content.

```
[root@ip-172-31-33-213 html]# ls -ld /var/www/html/
drwxr-xr-x. 6 root root 16384 May 11 15:17 /var/www/html/
[root@ip-172-31-33-213 html]# chown apache /var/www/html
[root@ip-172-31-33-213 html]# ls -ld /var/www/html
drwxr-xr-x. 6 apache root 16384 May 11 15:17 /var/www/html
[root@ip-172-31-33-213 html]#
```

```
[root@ip-172-31-33-213 html]# id apache
uid=48(apache) gid=48(apache) groups=48(apache)
[root@ip-172-31-33-213 html]# chown -R apache *
[root@ip-172-31-33-213 html]#
```
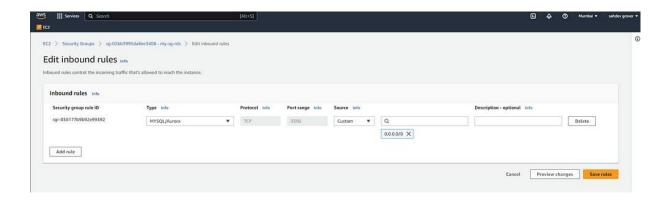
Step 6: Download Php for running WordPress.

Write this command to install the php7.2 version

```
[root@ip-172-31-33-213 html]# amazon-linux-extras install php7.2
```

```
Topic php7.2 has end-of-support date of 2020-11-30
Installing php-pdo, php-fpm, php-mysqlnd, php-cli, php-json
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-php7.2
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

Step 7: Edit inbound rules which enable tra c to the RDS instance.


```
[root@ip-172-31-33-213 html]# systemctl restart httpd
```

Then restart the httpd service by using the command given below.

Now you can access the WordPress installation page by navigating to your EC2 instance's public IP address in a web browser.
Follow the on-screen instructions to complete the WordPress installation.

ENTER YOUR DATABASE CONNECTION DETAILS

THIS, WE HAVE SEEN ABOVE

CONCLUSION

Congratulations on successfully deploying your WordPress website on AWS!

Congratulations on successfully deploying your WordPress website on AWS!