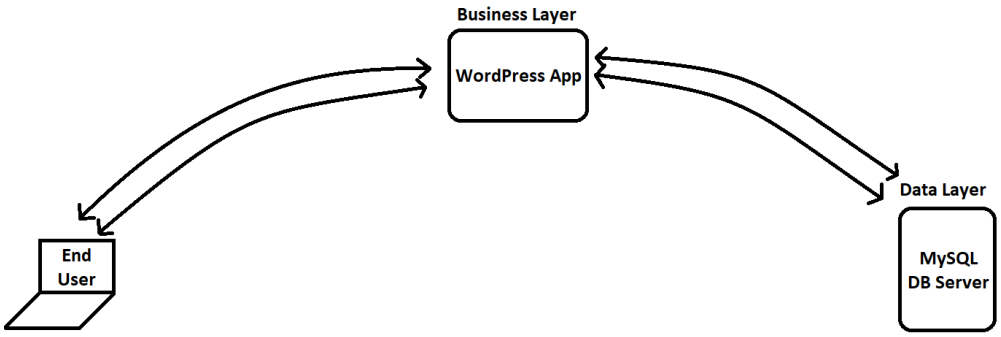
AWS Session 13

Summary – 09-03-2023

- A **multi-tier application** is a software application that is divided into multiple tiers or layers. Each tier is responsible for specific functions and interacts with other tiers to provide the end user with a complete application.

- **Example of 3–Tier Application**:



- **WordPress** is a blogging website. For deploying websites, we have multiple ways in AWS such as **EC2 instance, Lambda function & EKS**. But we deploy WordPress using the **Apache httpd** package in the EC2 instance.

- For managing database service there are 2 ways such as **self-managed and AWS-managed**. In self-managed services, the customer manages and monitors their infrastructure, applications, and databases. But in AWS-managed services, AWS manages & monitors infrastructure, applications, and databases.

- AWS provides **DBaaS** which is **Amazon Relational Database Service (RDS)**. It is a managed database service. RDS supports multiple popular

database engines such as MySQL, PostgreSQL, Oracle, SQL Server, MariaDB, and Amazon Aurora.

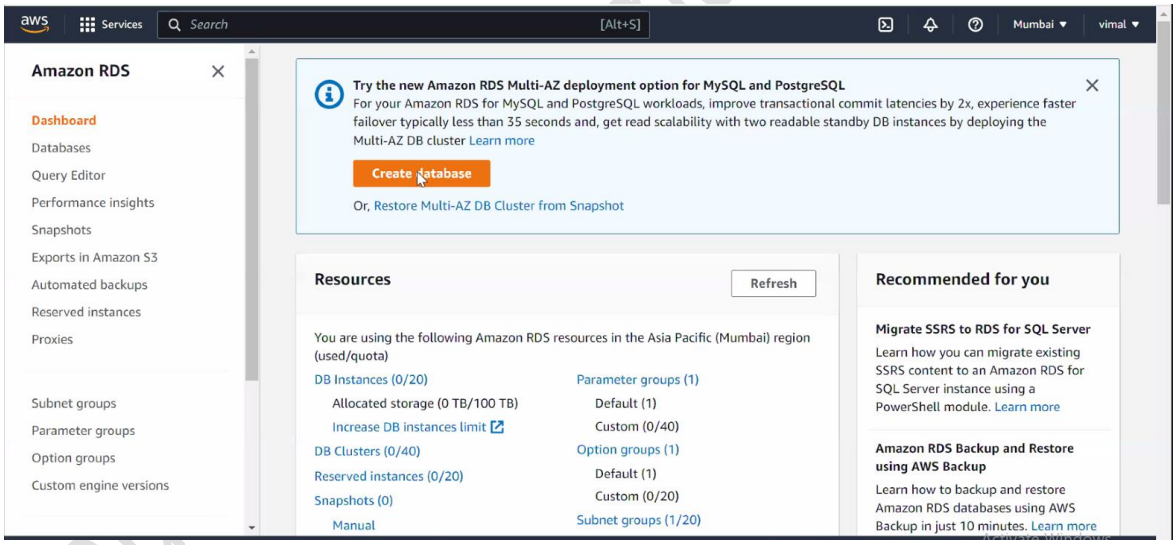
- One of the main benefits of using RDS is its ease of use. With RDS, users can launch a fully functional relational database with just a few clicks from the AWS Management Console.

- **Deploy the 3-tier application in AWS**:

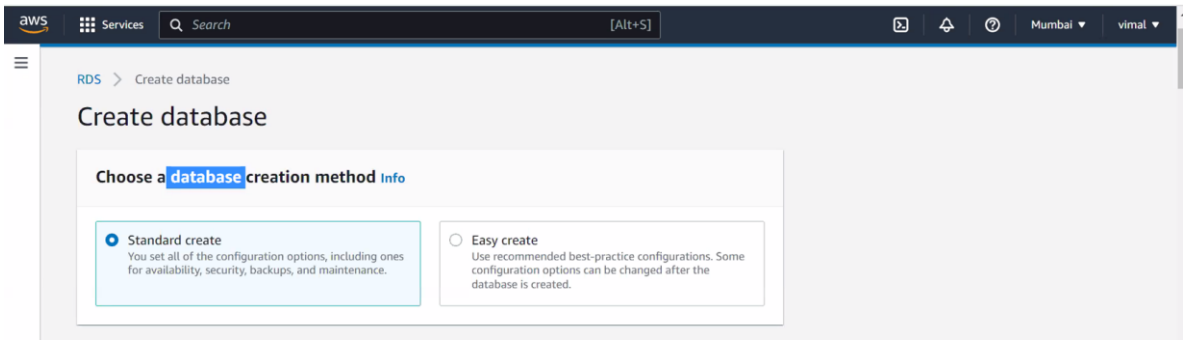
  1) Deploy DB Service using RDS.
  2) Deploy a WordPress website on an Amazon EC2 instance.

- **Deploy DB Service using RDS**:

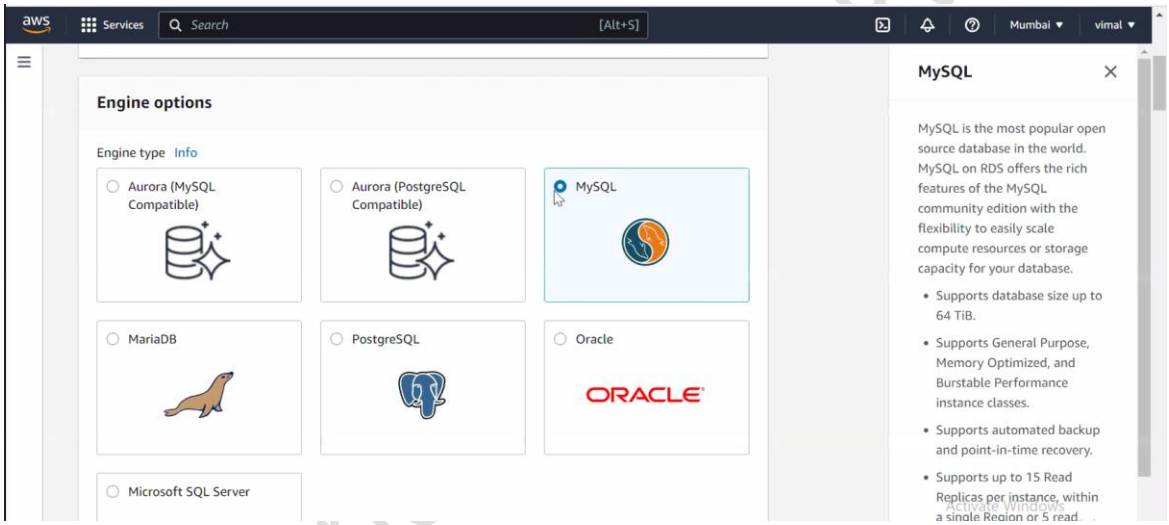
Step 1: Go to RDS and click on “Create database”.



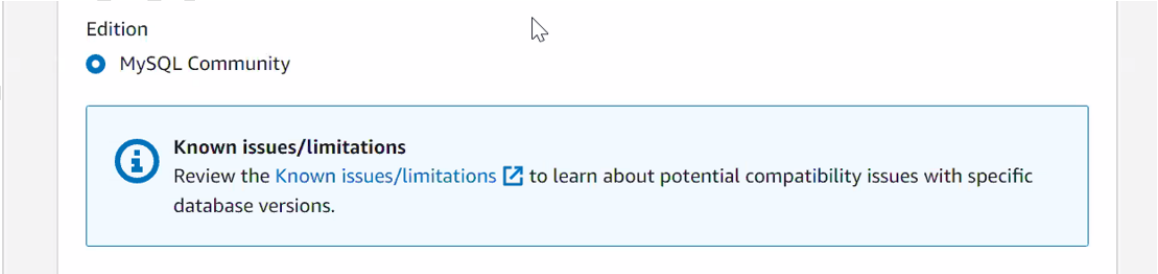
Step 2: Choose a database creation method.



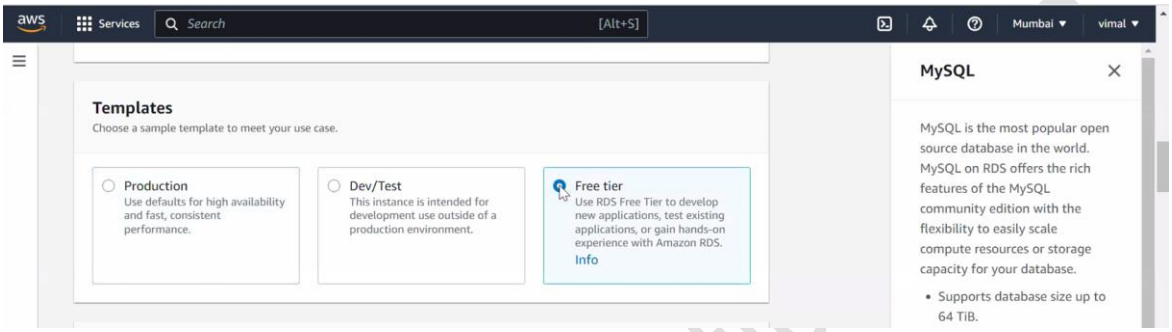
Step 3: Choose the database engine you want to use (e.g., MySQL, PostgreSQL, Oracle, SQL Server, MariaDB, or Amazon Aurora).



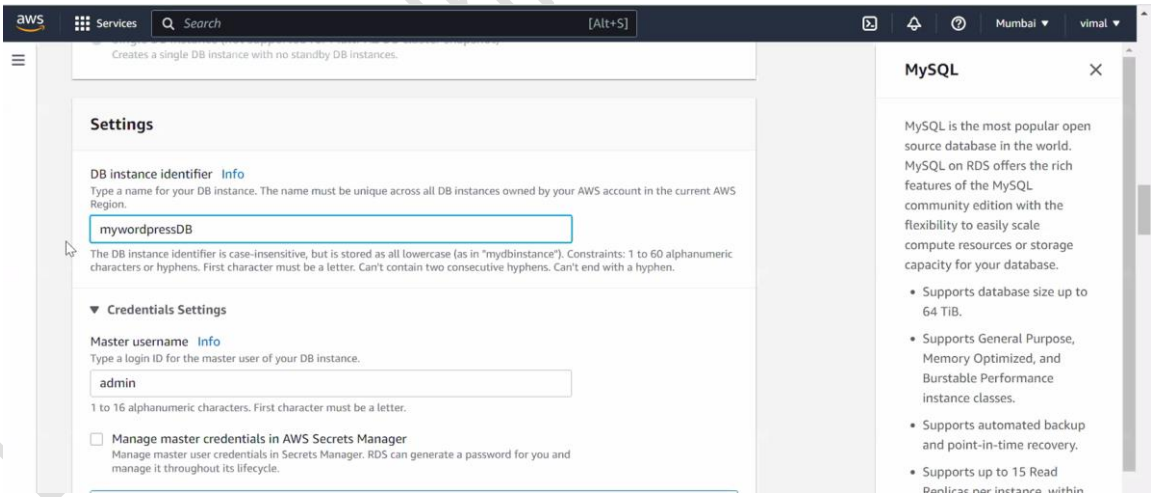
Step 4: Select the version and edition of the database engine you want to use.

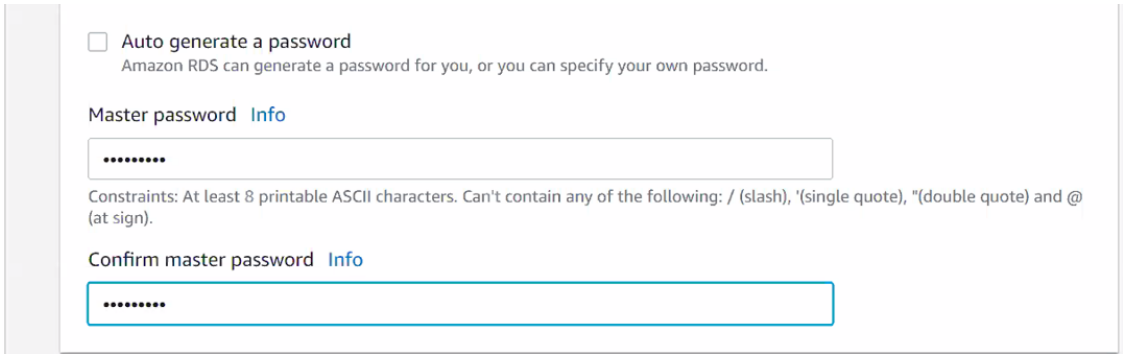


Step 5: Select the templates you want to use.

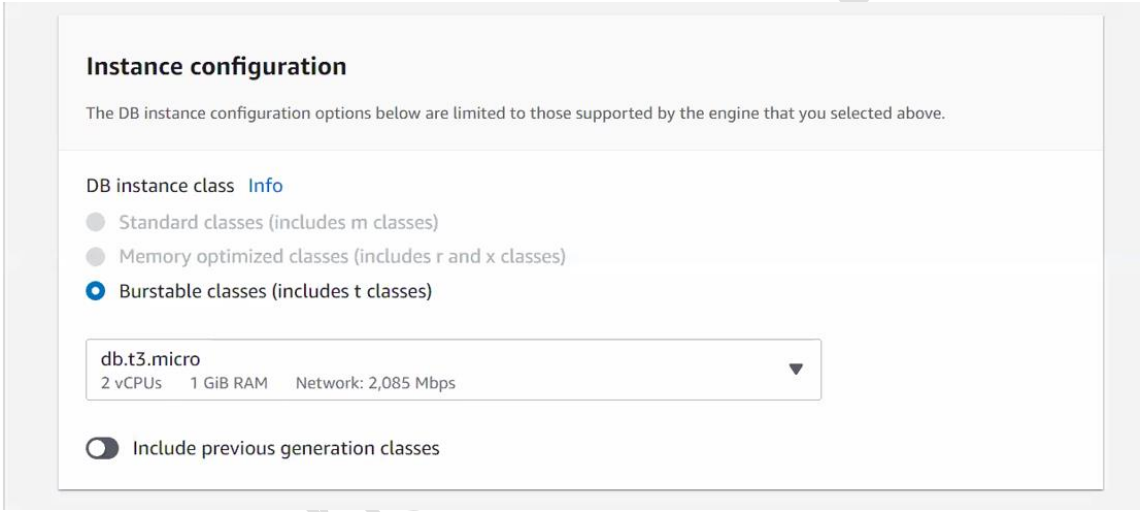


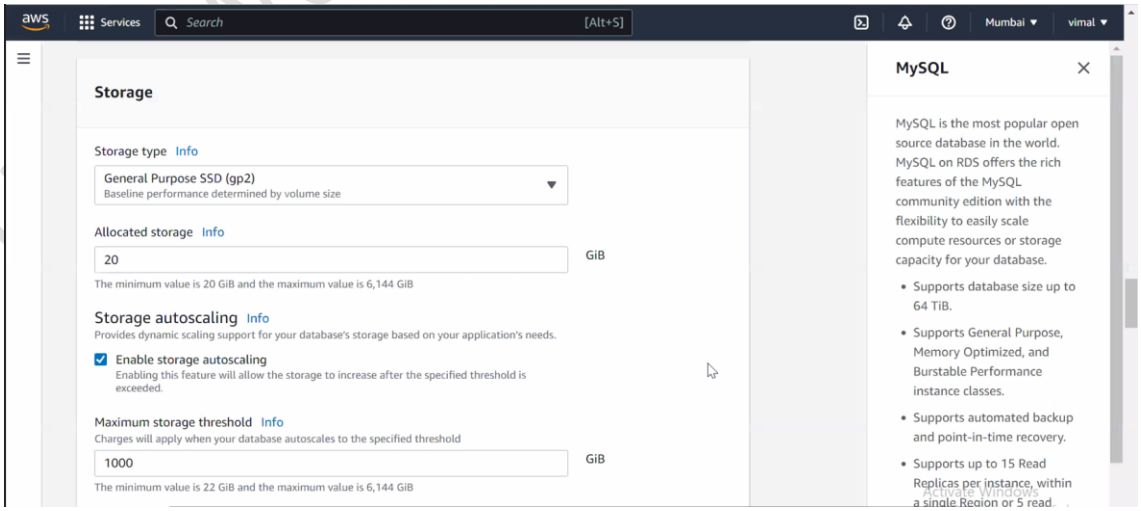
Step 6: Configure the database settings such as the database name, username, and password.



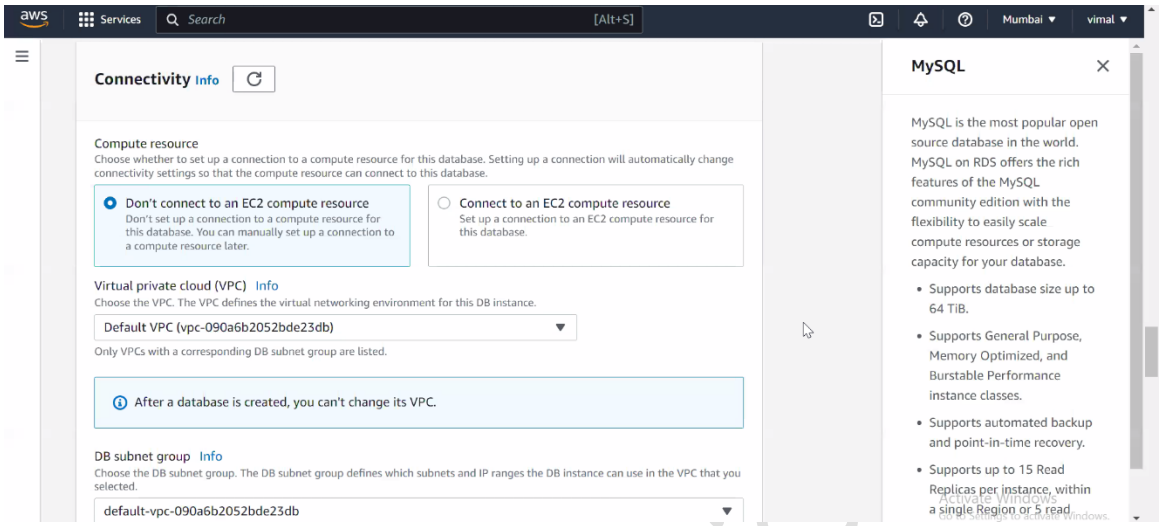


Step 7: Choose the appropriate instance class based on your performance and storage needs.

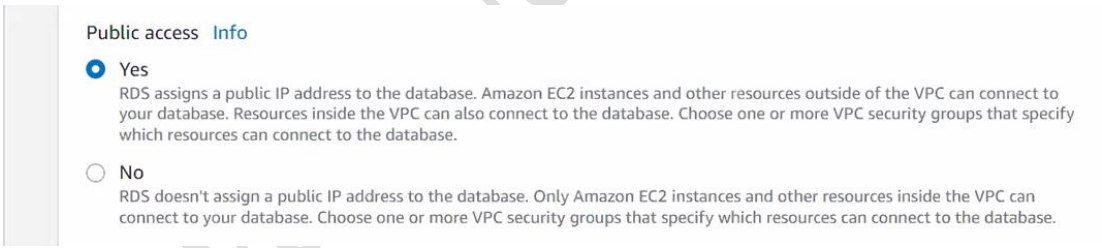


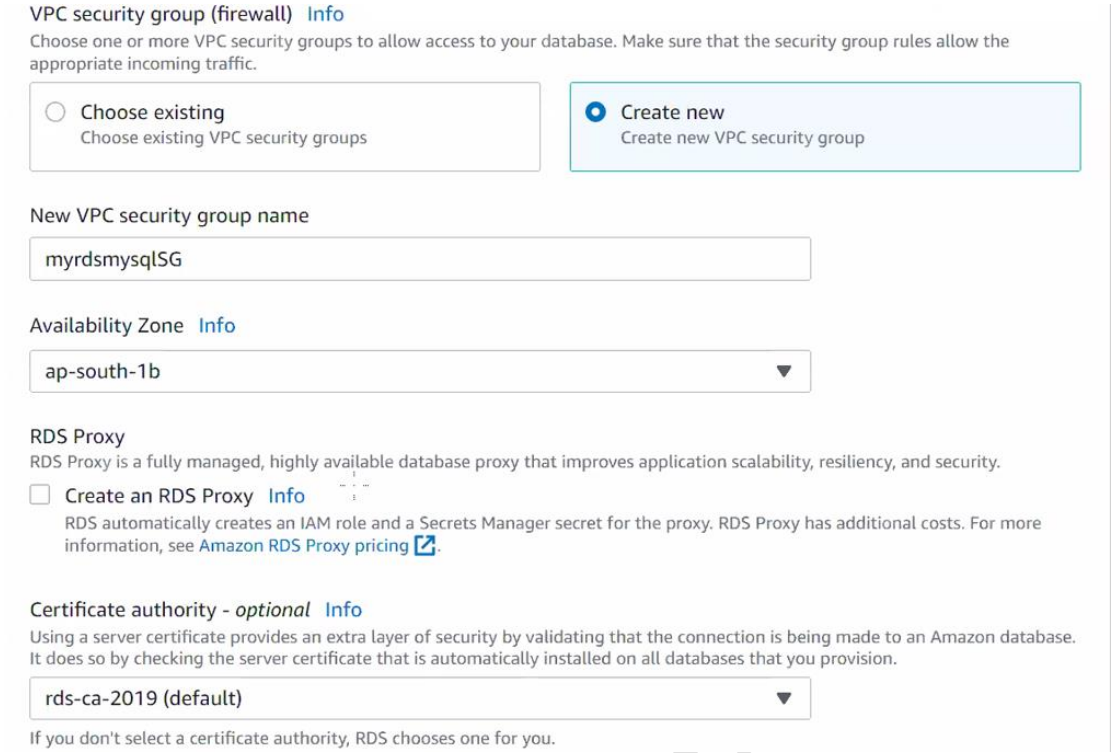


Step 8: Select the appropriate VPC and subnet group to launch the database in.

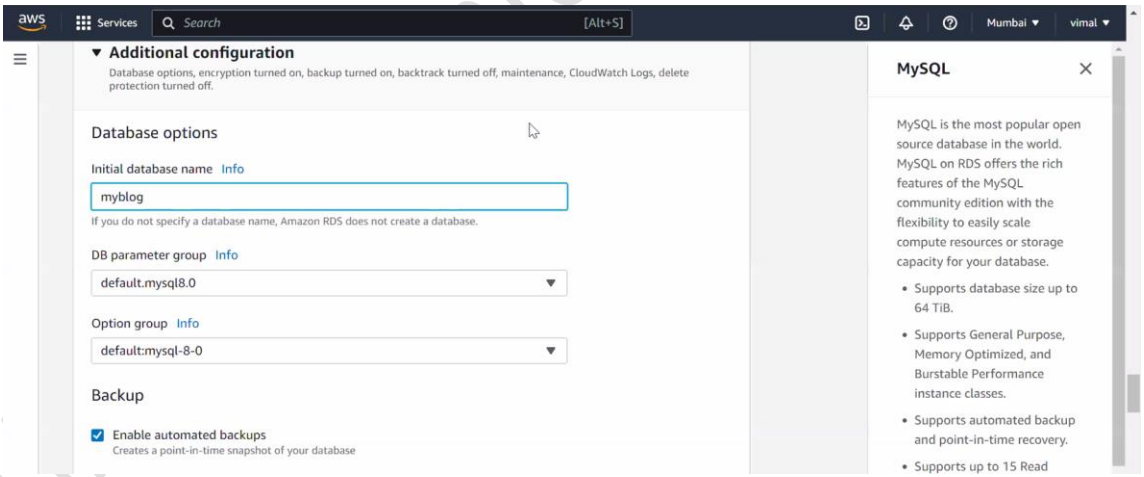


Step 9: Configure the database security group to control access to the database.





Step 10: Create the initial database.



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

- **Deploy a WordPress website on an Amazon EC2 instance**.

Step 1: Launch an EC2 instance using Amazon Linux AMI.

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

 # yum install httpd -y

 # systemctl enable httpd --now

Step 3: Go to the Apache web server web page location and download WordPress and extract it.

```
[root@ip-172-31-15-120 ~]# cd /var/www/html/
[root@ip-172-31-15-120 html]# ls
[root@ip-172-31-15-120 html]# pwd
/var/www/html
[root@ip-172-31-15-120 html]# wget https://wordpress.org/latest.tar.gz
--2023-03-09 16:50:45--  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...
```

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

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

```
[root@ip-172-31-15-120 html]# ls
latest.tar.gz  wordpress
[root@ip-172-31-15-120 html]# cp  -r wordpress/*    .
[root@ip-172-31-15-120 html]# pwd
/var/www/html
[root@ip-172-31-15-120 html]# ls
index.php       readme.html      wp-admin             wp-config-sample.php  wp-includes       wp-login.php     wp-signup.php
latest.tar.gz   wordpress        wp-blog-header.php   wp-content            wp-links-opml.php  wp-mail.php      wp-trackback.php
license.txt     wp-activate.php  wp-comments-post.php  wp-cron.php          wp-load.php       wp-settings.php  xmlrpc.php
```

Step 5: Make Apache owner for all WordPress content.

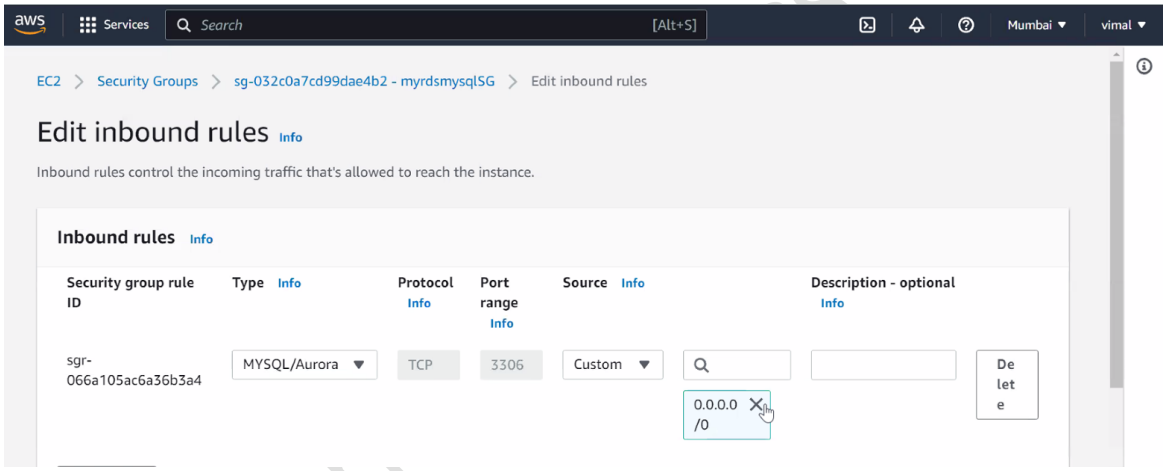
```
[root@ip-172-31-15-120 html]# ls -ld /var/www/html/
drwxr-xr-x 5 root root 4096 Mar  9 16:52 /var/www/html/
[root@ip-172-31-15-120 html]# chown  apache /var/www/html/
[root@ip-172-31-15-120 html]# ls -ld /var/www/html/
drwxr-xr-x 5 apache root 4096 Mar  9 16:52 /var/www/html/
```

```
[root@ip-172-31-15-120 html]#
[root@ip-172-31-15-120 html]# id  apache
uid=48(apache) gid=48(apache) groups=48(apache)
[root@ip-172-31-15-120 html]# chown  -R  apache  *
```

Step 6: Download php for running WordPress.

```
[root@ip-172-31-15-120 ~]# 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
```

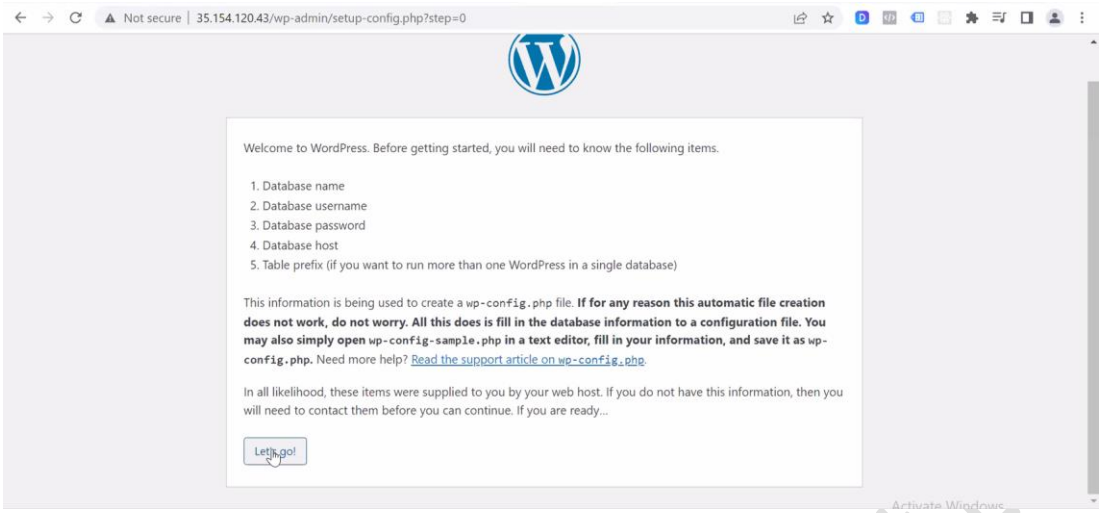
Step 6: Edit inbound rules which enable traffic to the RDS instance.

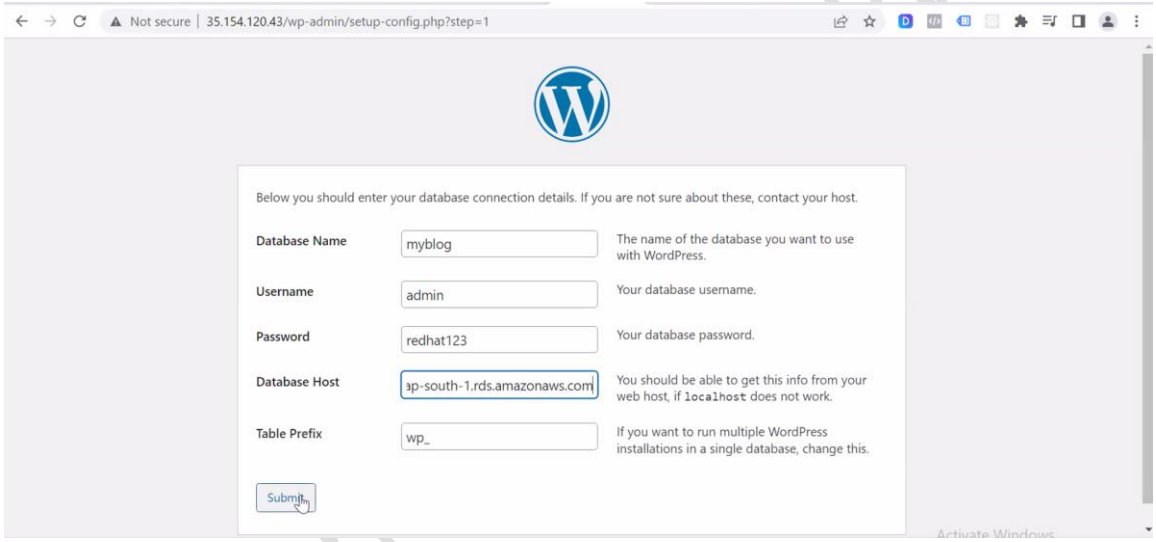


Then finally restart the httpd service.

 # systemctl restart httpd

- Now hit the IP of the WordPress instance and they ask about database information.


Not secure | 35.154.120.43/wp-admin/setup-config.php?step=0
Welcome to WordPress. Before getting started, you will need to know the following items.
1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)
This information is being used to create a wp-config.php file. If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php. Need more help? Read the support article on wp-config.php.
In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...
Let's go!


Not secure | 35.154.120.43/wp-admin/setup-config.php?step=1
Below you should enter your database connection details. If you are not sure about these, contact your host.
Database Name
myblog
The name of the database you want to use with WordPress.
Username
admin
Your database username.
Password
redhat123
Your database password.
Database Host
ap-south-1.rds.amazonaws.com
You should be able to get this info from your web host, if localhost does not work.
Table Prefix
wp_
If you want to run multiple WordPress installations in a single database, change this.
Submit


Not secure | 35.154.120.43/wp-admin/setup-config.php?step=2
All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to...
Run the installation