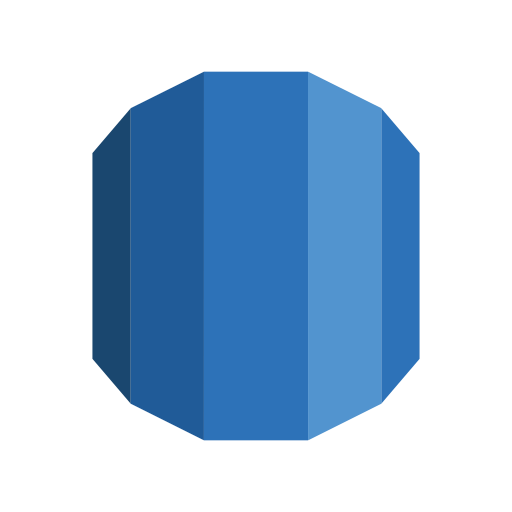
AWS USE CASE – 4

Deploying WordPress with AWS RDS as Backend

In this use case, we will find how to launch a WordPress application on the top of the ec2 instance on AWS cloud and how to integrate it with the Relational Database Service (RDS) of AWS Cloud.

**AWS RDS (Relational Database Service)**



Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

* You can use database engines are: IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle Database, and PostgreSQL.
* Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
* You can turn on automated backups, or manually create your own backup snapshots.
* In RDS, security in your database package, you can control access by using AWS Identity and Access Management (IAM) to define users and permissions.

**WordPress**

WordPress is a popular open-source content management system (CMS) used to create and manage websites. It's known for its flexibility, user-friendly interface, and a vast ecosystem of plugins and themes.

**Task Details**

**1. Create an AWS EC2 instance**

launch an EC2 instance (WordPress-Server) on the AWS cloud with linux 2 AMI and allow ssh (port 22) and http (port 80) from internet.

**2. Configure Apache Webserver and WordPress on EC2**

Connect SSH to the instance running on AWS cloud from terminal. After connecting to the instance, we require php7.3 and httpd (Apache Webserver) software.

* **To Install the Apache(httpd)**

*sudo yum install httpd -y*

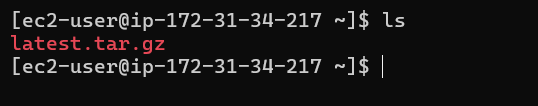
* **To install php7.3**

*sudo amazon-linux-extras install php7.3 -y*

***Note: that not every version of PHP is not compatible with all the versions of the Database.***

* **To download WordPress**

*wget*[*https://wordpress.org/latest.tar.gz*](https://wordpress.org/latest.tar.gz)

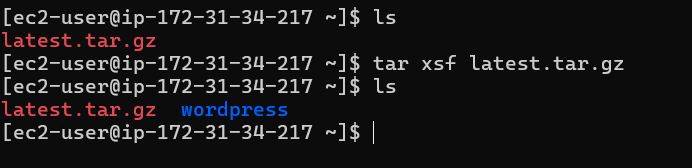


After download the WordPress we can see that the “latest.tar.gz” one compressed package is downloaded.

So, we need to extract this.

* **To extract the downloaded package**

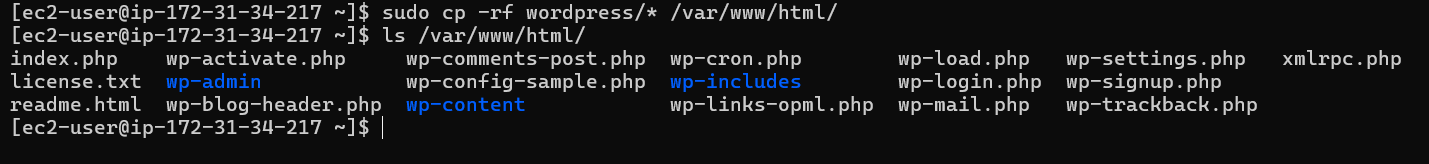
*tar -xzf latest.tar.gz*



After extracting, a new directory named ‘**wordpress**’ will be created.

* **Now, we have to copy the entire content of this folder to the default document root of the Apache WebServer i.e., to the /var/www/html directory.**

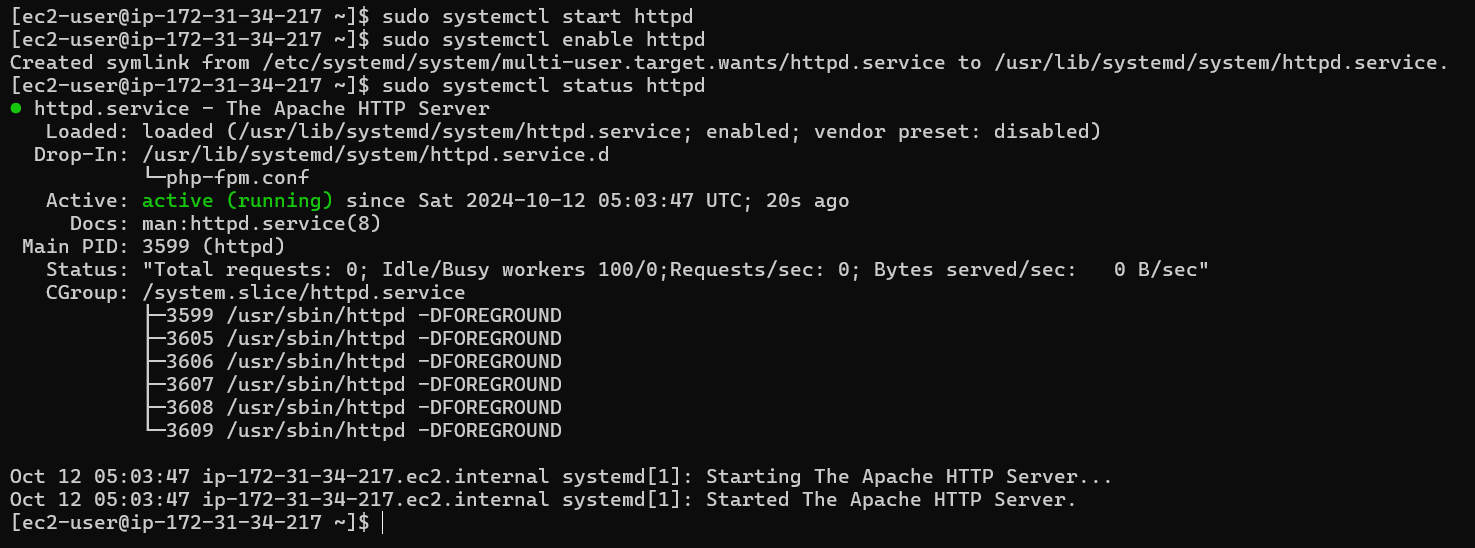
*cp -rf wordpress/\* /var/www/html/*



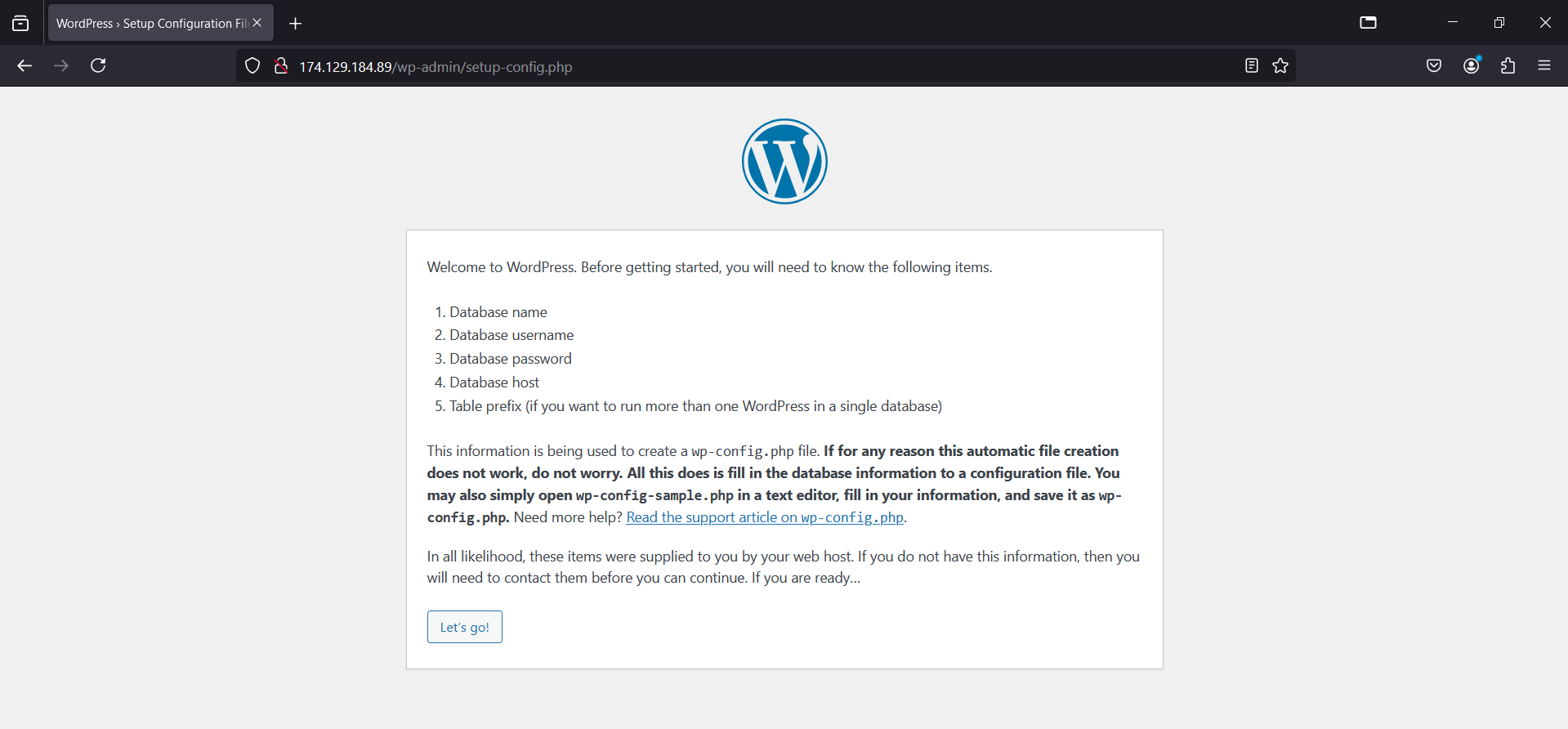
* **To start the httpd**

*systemctl start httpd*

*systemctl enable httpd*



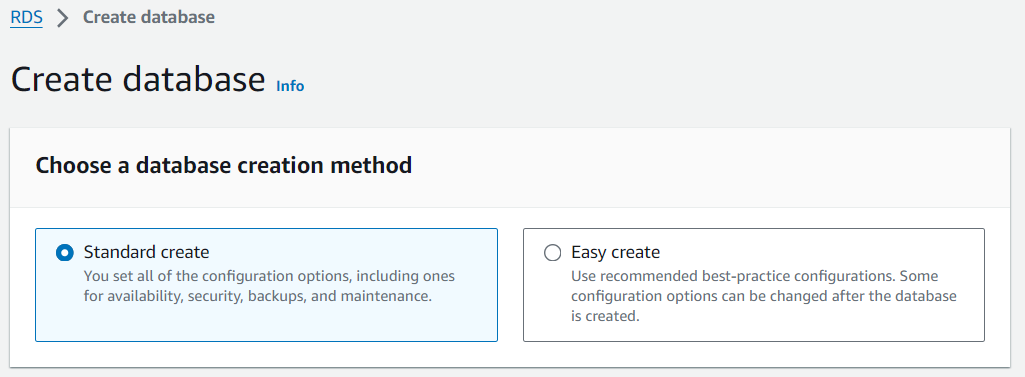
And now if try to access the web server, it will ask for the details of the database server:



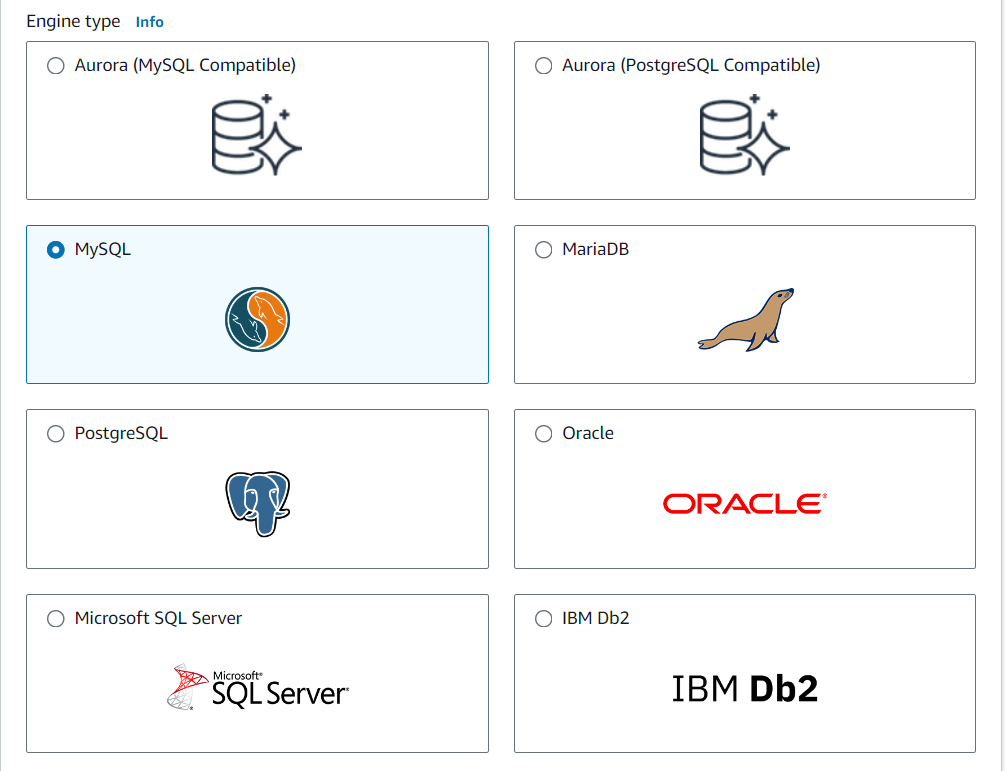
So, let us now set up the database for this WordPress. And we will use the RDS service of AWS to set up our database server.

**3. Create RDS Database**

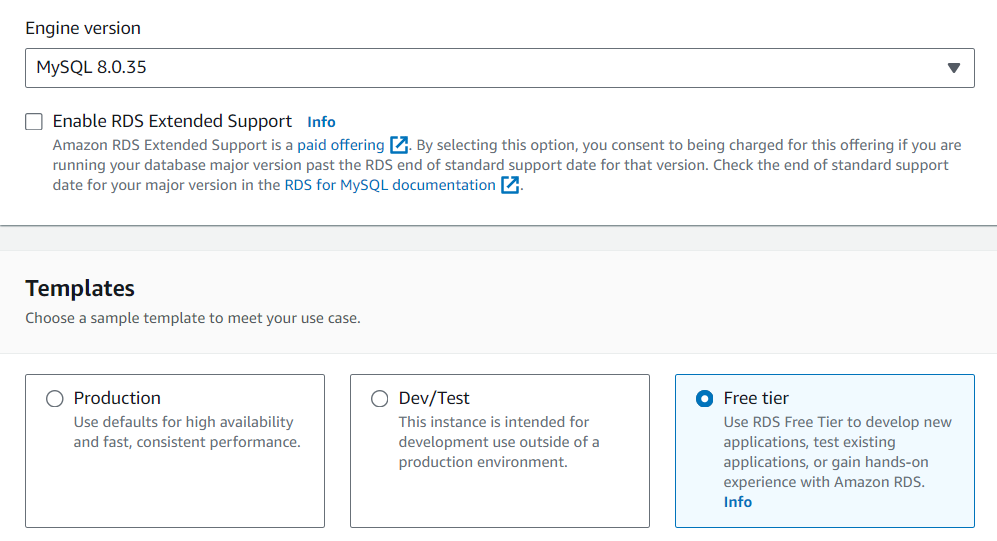
* Navigate to the RDS in AWS console.
* Click on **Create Database**.
* Choose the standard Database create option



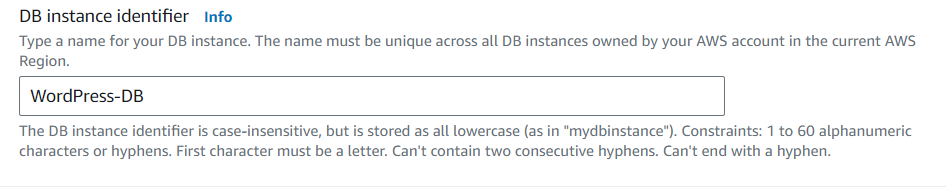
* Choose MySQL Database Engine

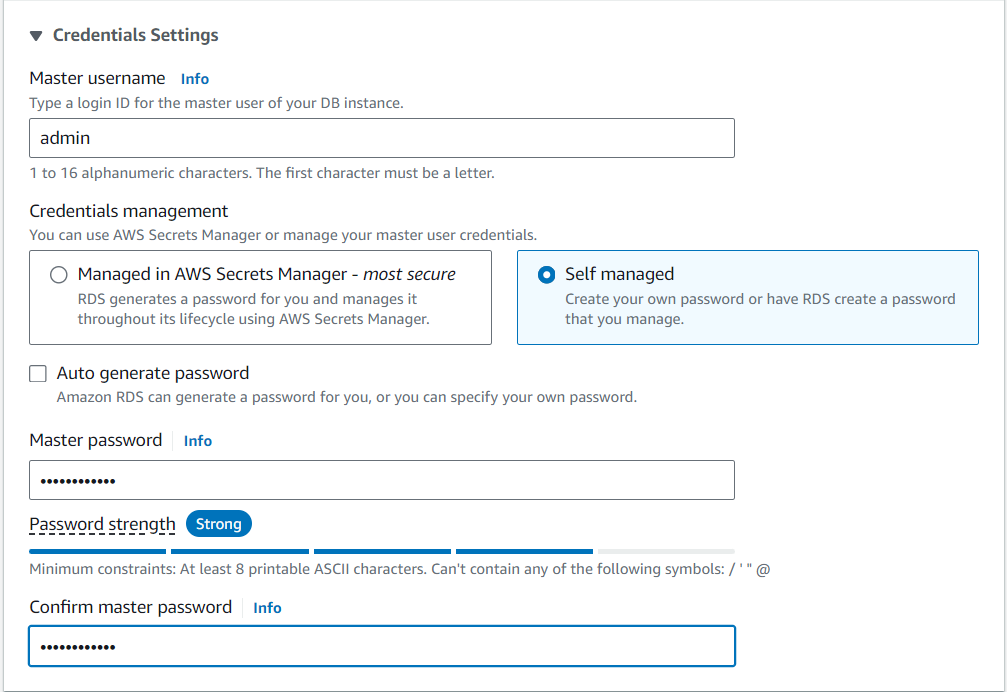


* Choose the engine version as MySQL 8.0.35 and Free tier template.

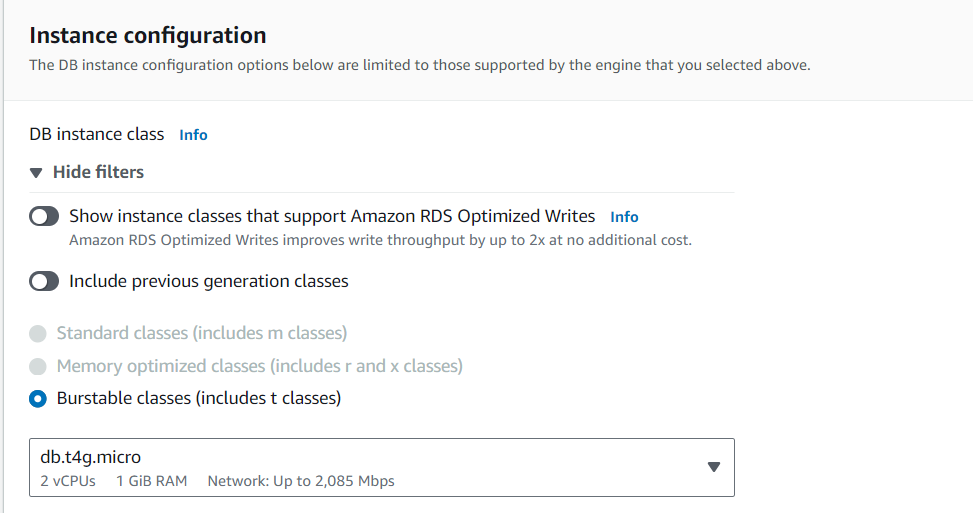


* Give the database instance name, master username and the master password.

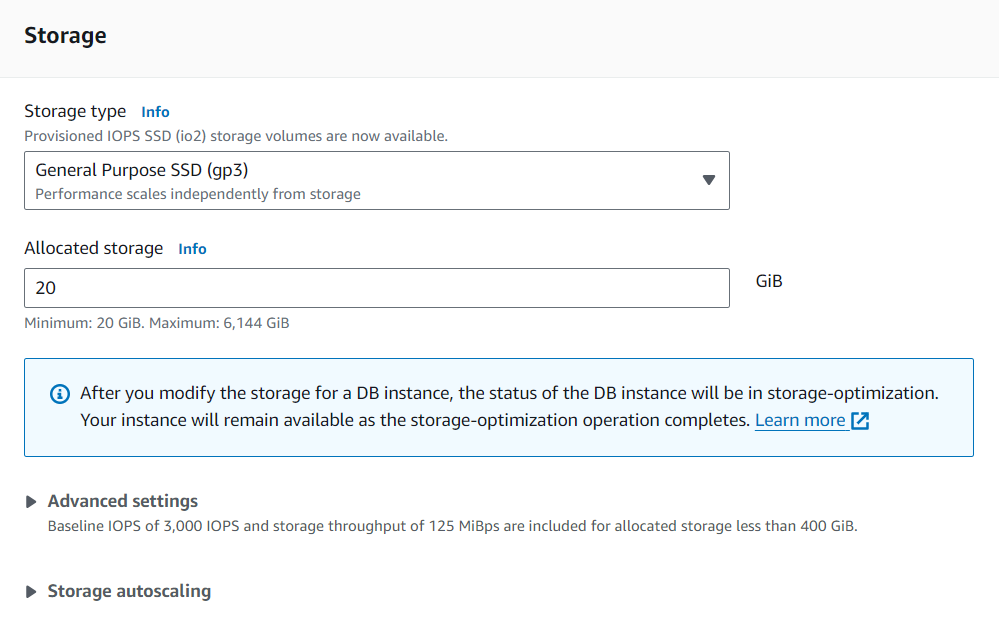




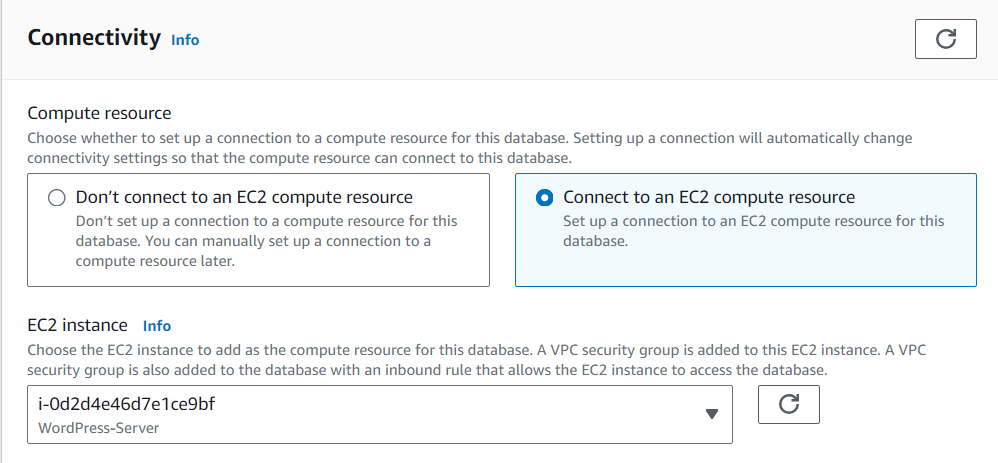
* Choose the desired DB Instance class.



* Keep default in storage option, as it’s enough for our config.

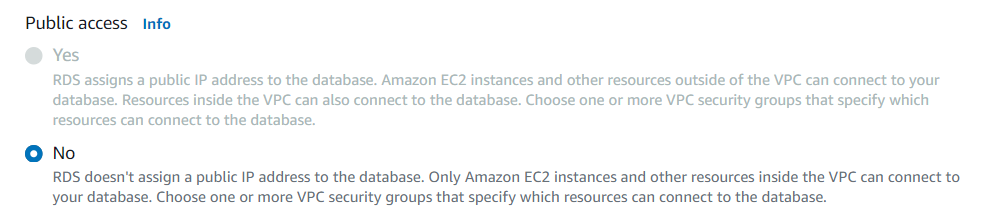


* Under Connectivity choose; **Connect to an EC2 compute resources**. And choose our WordPress-Server instance as the EC2 compute resource.



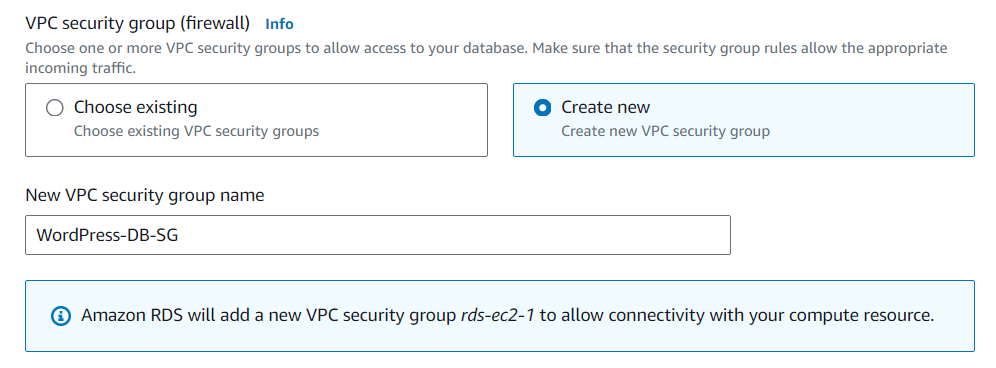
**Note: Here a VPC security group is added to the EC2 instance. And a VPC security group is also added to the database with an inbound rule that allows the EC2 instance access to the database.**

* Keep all the other setting as they are, we will be using default VPC, subnets.
* Here we disable the public access to the Database.

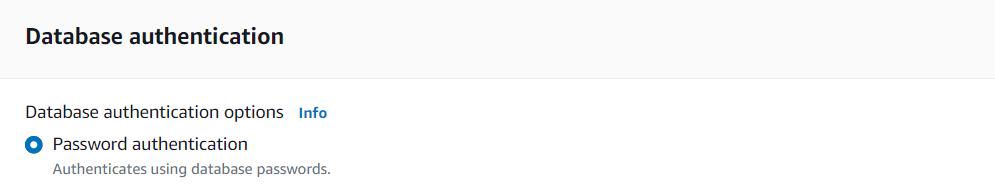


**Note: If you give the public access to the database, a public IP will be assigned to it and you can connect to the database from anywhere in the world. But if you are accessing this database from one of the amazon instances only, you can decide not to give it public access. By default, all the amazon instances will have internal connectivity in the same VPC.**

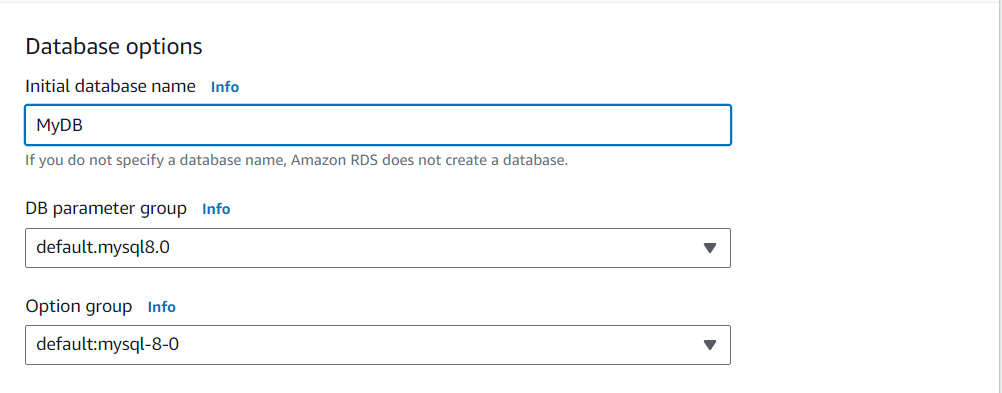
* Create another security group; Amazon RDS will add a new VPC security group *rds-ec2-1* to allow connectivity with your compute resource

****

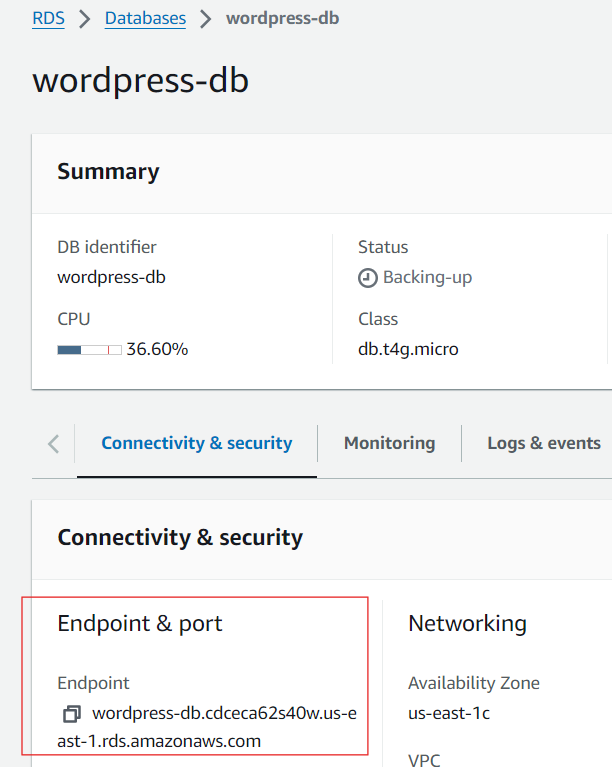
* Under Database Authentication choose password authentication.

****

* In the additional configuration section, create a database by providing a name to it.

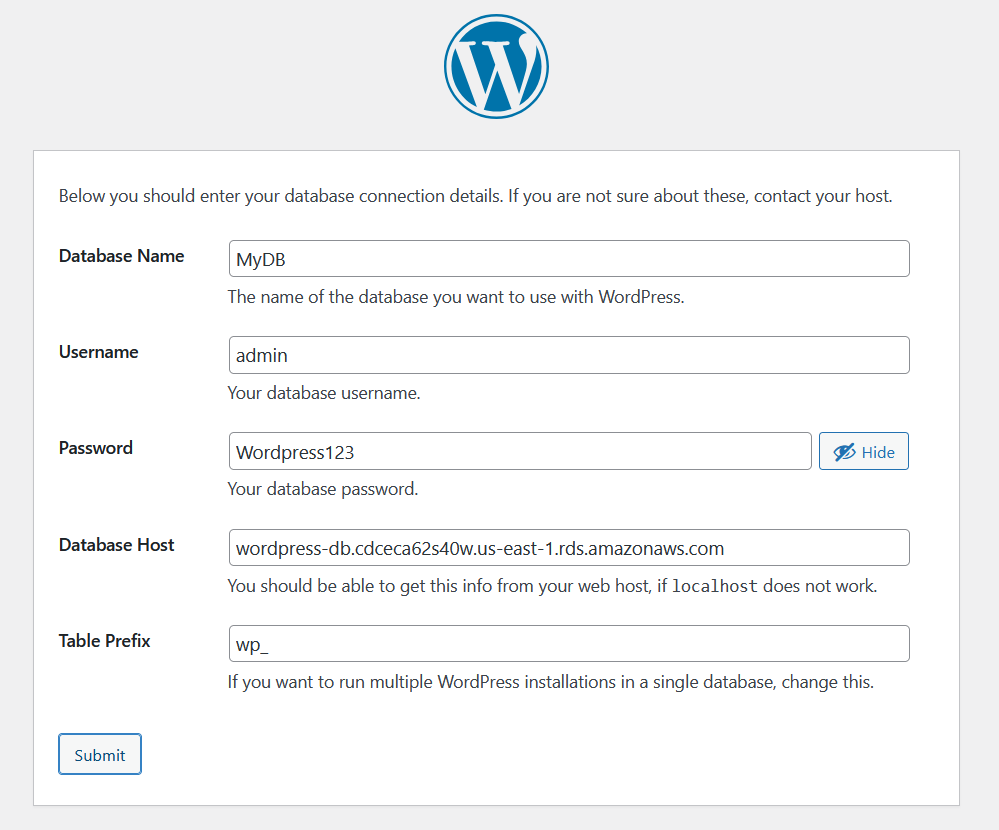


* Keep all the other settings as default And the RDS configuration has been finished. Click on create the database and wait for the process to finish. Once the database is created, click on the database and it will give an endpoint. We will use this endpoint to connect our WordPress with the database server.



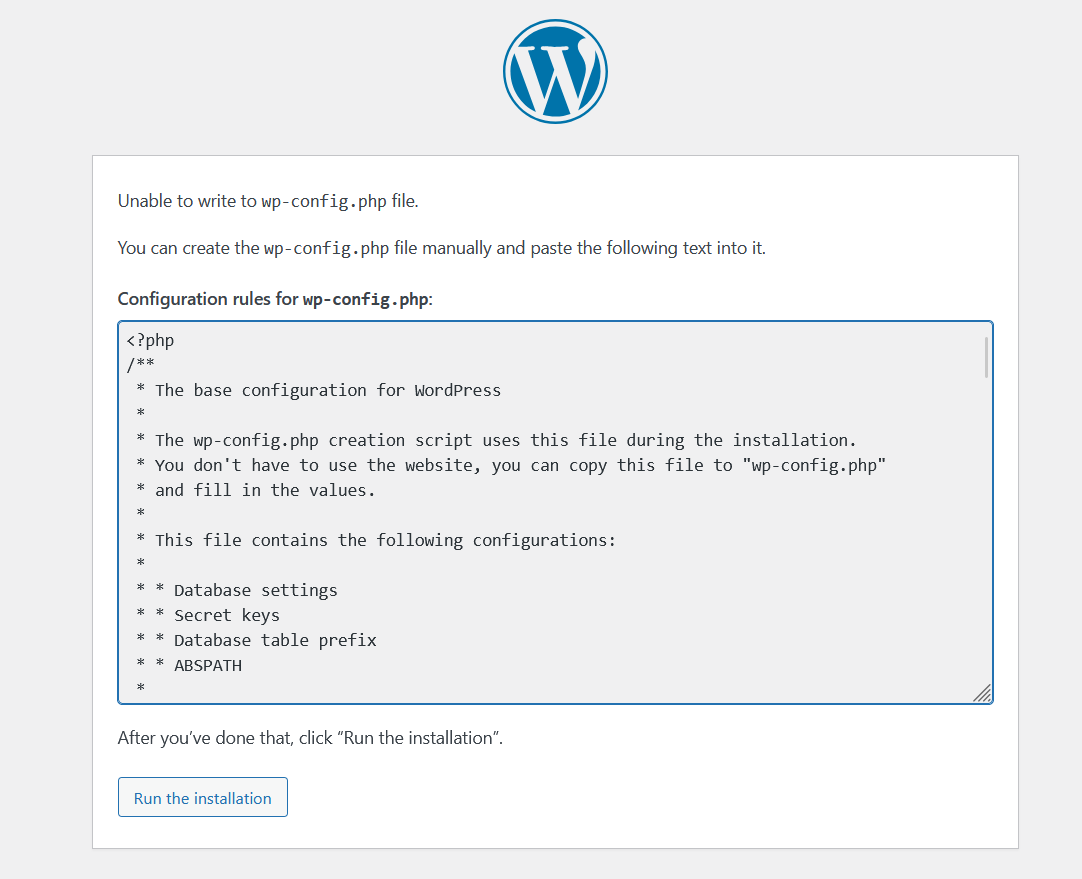
And now, we have to provide these database details to out WordPress:

Go to the WordPress URL and click on “Let’s go!”, and on the next page, provide the database details such as database name, username, password, and database host i.e., Endpoint.



And then, click on Submit.

Now, here comes the interesting part. On the next page, you will get a screen like this:



It says that it was unable to write to the wp-config.php file and it asks us to create the wp-config.php file in /var/www/html/ location and enter the given content in that file.

*Why did this error come and what does it mean?* Let me try to clear this out.

***Whatever database details we entered just now, we need to keep this information in some file so that whenever our WordPress wants to connect to the database, it can pick the information from that file. And here, WordPress uses wp-config.php named file to store and pick information.***

Now, why WordPress did not create this file and why it asked us to create the file on its behalf?

***Apache Webserver creates and uses an Apache user behind the scene to do certain things. Apache is a service user and it is the one that has to copy the code inside the wp-config file at /var/www/html location by creating it first. But here, we have not given any permissions to the apache user to do any changes in the /var/www/html location. Because it is not always recommended to give extra permissions to a service user.***

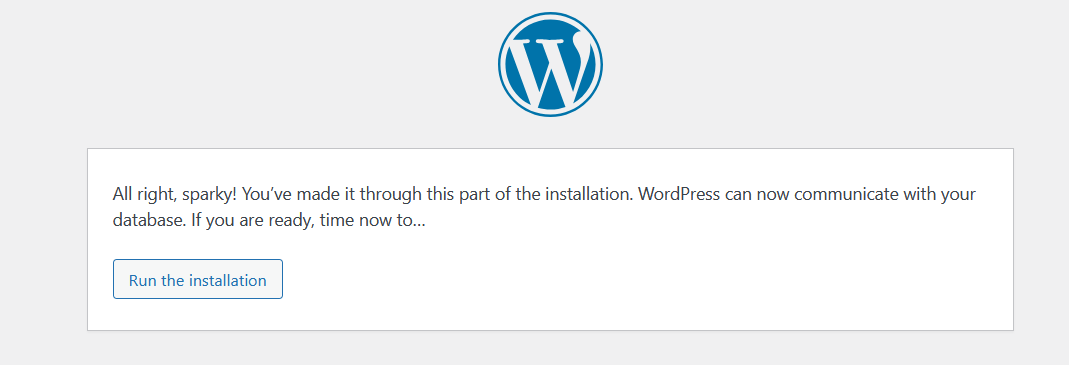
***But still, if we want that the configuration process should be done by Apache user and we do not have to do any manual configuration, here we have to provide the required permissions to the Apache user.***

Use the following commands to do so:

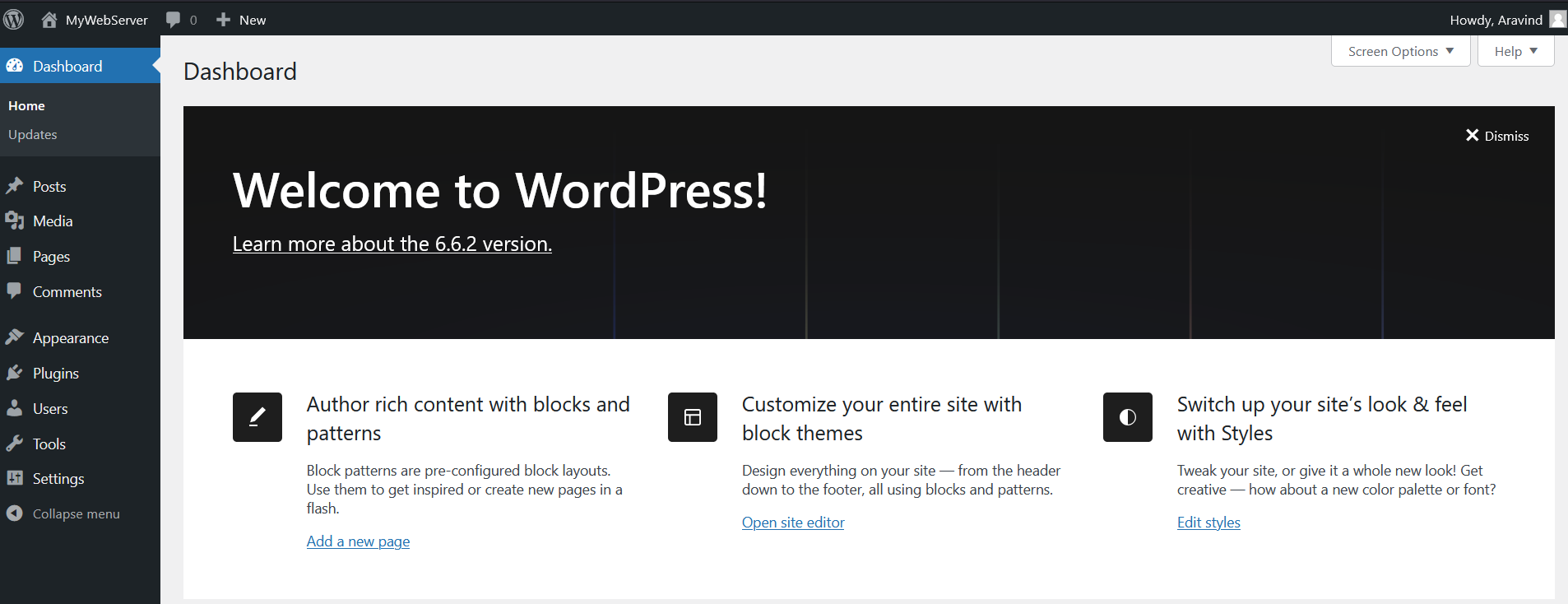
chown -R apache:apache /var/www/html/

After running the above commands, go back, again enter the database details and click on Submit.

And this time you will see that the wp-config.php file creation and its configuration is done automatically.



Click on **‘Run the Installation’**, set up your profile by providing the personal details and after login, you are all set to use your own WordPress for creating exciting blogs.



And hence, all the requirements of the task have been fulfilled.

Ref: https://medium.com/@shubham134/integrating-wordpress-with-rds-on-aws-cloud-7f136588e35d