Deploy WordPress with Amazon RDS

Objective: Create a fully managed MySQL database and deploy a wordpress

## Module 1: Creating a MySQL Database with RDS

In this module, you will create a MySQL database for your WordPress site.

### Implementation:

### Create MySQL Database

### 

### Click on MySQL

### 

### Click on Free tier

### 

### 

### User: admin

### Passw: admin123456

### 

### 

### 

### 

### 

### Database is being created

### 

## Module 2: Creating an EC2 Instance

In this module, you will create an Amazon EC2 instance to run your WordPress site.

### Objective: Create a EC@ instance to run wordpress site

### Choose an Amazon Machine image

### 

### 

### Choose an instance type

### 

### Configuring a Security Group

### SSH traffic from your current IP address so you can use the SSH protocol to log into the EC2 instance and configure WordPress

### HTTP traffic from all IP addresses so users can view WordPress site

### 

### Create new Security group

### 

### 

### 

### Launch and get SSH Key

### 

### 

### Launch Instance

### 

### 

## Module 3: Configuring Your RDS Database

In this module, you will configure the RDS database to allow access to specific entities.

### Objective: In this module, we will configure the RDS database to allow access to specific entities.

### Implementation:

### Allow EC2 instance to access RDS Database

### 

### Click on Security Group

### 

### Edit Security Group

### 

### 

### Select Wordpress security Group

### 

### 

### 

### SSH into your EC2 Instance

### 

### 

### Create a Database User

First, run the following command in your terminal to install a MySQL client to interact with the database.

sudo yum install -y mysql

### 

### 

### Next, find the hostname for your RDS database in the AWS console. In the details of your RDS database, the hostname will be shown as the Endpoint in the Connectivity & security section.

### 

### set an environment variable for your MySQL host.

### export MYSQL\_HOST=<your-endpoint>

### export MYSQL\_HOST = wordpress.cqcelh3nbfkk.us-east-2.rds.amazonaws.com

### 

### Error

### 

### Remove the spaces between = so it might not give an error

### 

### Successful

### 

Next, run the following command in your terminal to connect to your MySQL database. Replace “<user>” and “<password>” with the master username and password you configured when creating your RDS database.

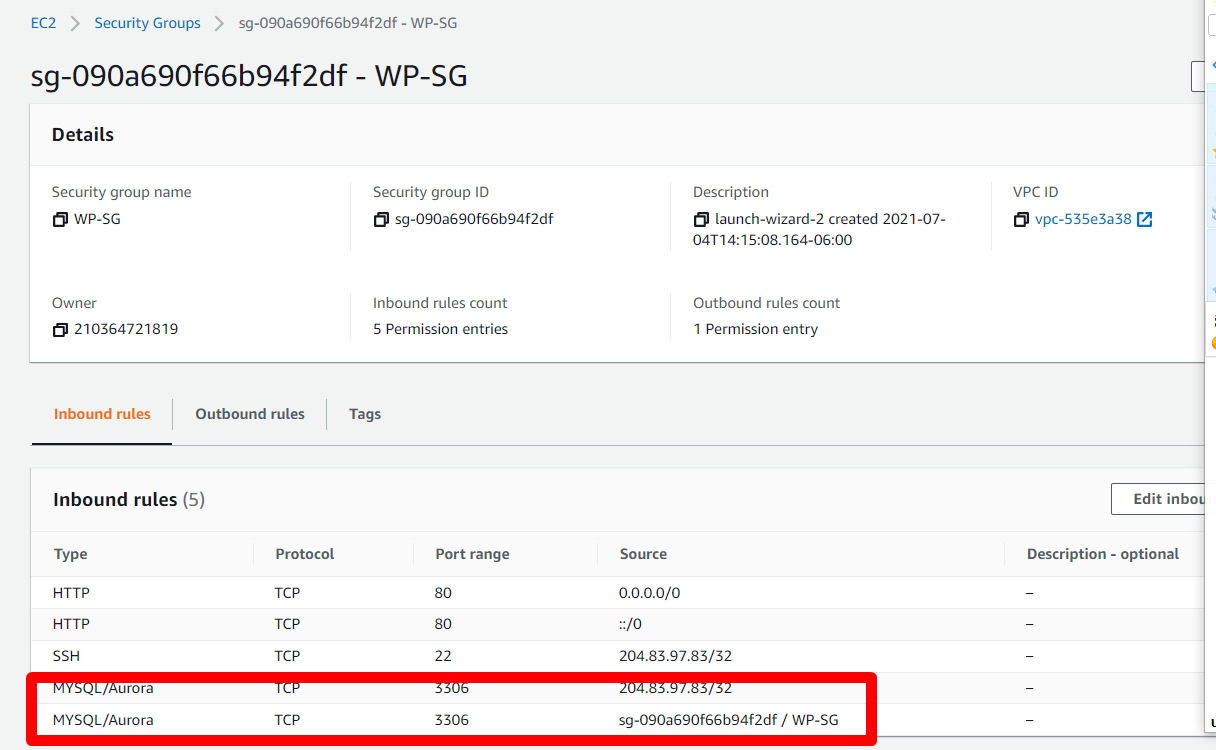
mysql --user=<user> --password=<password> wordpress

mysql –-user=admin -–password=test123456 wordpress

I was getting an error on connecting mysql from EC2 instance and it had to dig a little bit in deep.

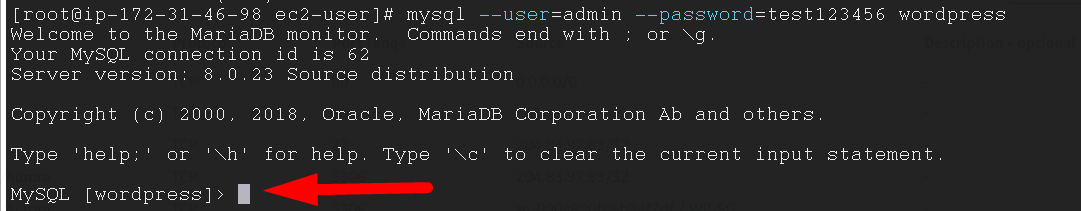
It had to check the inbound rules for MYSQl/ Aurora

You need to allow a source as security group which is attached to the MYSQL db instance on port 3306



If you connected successfully, your terminal should indicate connection to the MySQL database as shown in the following image

Connection Successful to RDS Db instance



create a database user for your WordPress application and give it permission to access the “wordpress” database.

Run the following commands in your terminal:

CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';

GRANT ALL PRIVILEGES ON wordpress.\* TO wordpress;

FLUSH PRIVILEGES;

Exit

CREATE USER 'wordpress' IDENTIFIED BY 'adminwordpress';

GRANT ALL PRIVILEGES ON wordpress.\* TO wordpress;

FLUSH PRIVILEGES;

Exit

### 

### Your EC2 instance now has network access to your RDS database. Further, you created a database user to be used by your Wordpress application.

## Module 4: Configuring WordPress on EC2

In this module, you will finish up the work to make your WordPress site live.

### Objective: Install the WordPress application and dependencies on the EC2 instance

### Install Apache web server

### sudo yum install –y httpd

### 

### 

### Check status of apache

### Sudo systemctl status httpd

### 

### Start Apache web server

### Sudo systemctl start httpd

### 

### Check status of apache if it is started

### Sudo systemctl status httpd

### 

### Go to instance page and get the public ip of web server

### 

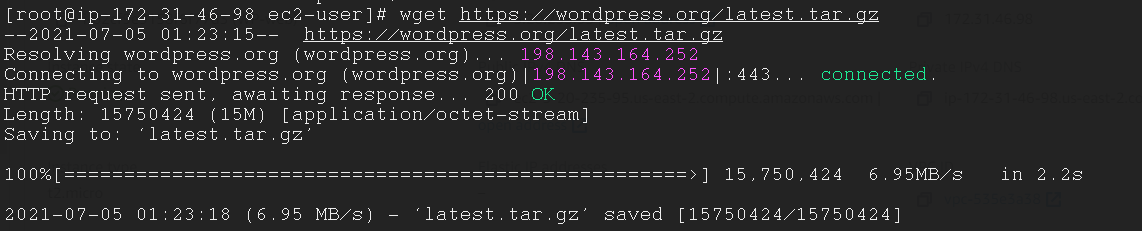
### Enter into browser and see if apache Test page is being displayed

### 

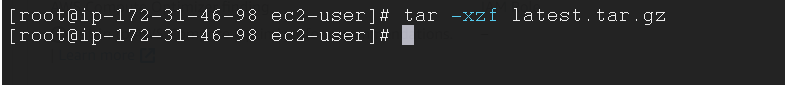
### Download and configure WordPress

First, download and uncompress the software by running the following commands in your terminal:

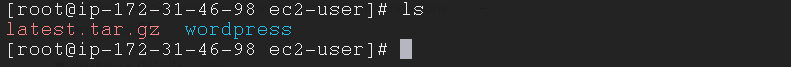
wget <https://wordpress.org/latest.tar.gz>



tar -xzf latest.tar.gz

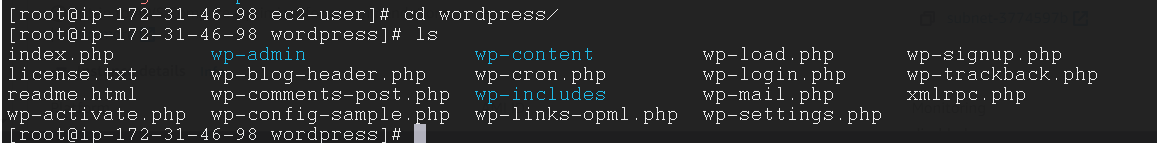


If you run “ls” to view the contents of your directory, you will see a tar file and a directory called wordpress with the uncompressed contents.



Change into the wordpress directory and create a copy of the default config file using the following commands:

Cd wordpress



Change into the wordpress directory and create a copy of the default config file using the following commands:

cp wp-config-sample.php wp-config.php

### 

### open the wp-config.php file using the [nano](https://www.nano-editor.org/" \t "_blank) editor by running the following command.

### Nano wp-config.php

### 

You need to edit two areas of configuration.

First, edit the database configuration by changing the following lines:

The values should be:

* DB\_NAME: “wordpress”
* DB\_USER: The name of the user you created in the database in the previous module
* DB\_PASSWORD: The password for the user you created in the previous module.
* DB\_HOST: The hostname of the database that you found in the previous module.

### 

The second configuration section you need to configure is the Authentication Unique Keys and Salts. It looks as follows in the configuration file:

/\*\*#@+

\* Authentication Unique Keys and Salts.

\*

\* Change these to different unique phrases!

\* You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}

\* You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.

\*

\* @since 2.6.0

\*/

define( 'AUTH\_KEY', 'put your unique phrase here' );

define( 'SECURE\_AUTH\_KEY', 'put your unique phrase here' );

define( 'LOGGED\_IN\_KEY', 'put your unique phrase here' );

define( 'NONCE\_KEY', 'put your unique phrase here' );

define( 'AUTH\_SALT', 'put your unique phrase here' );

define( 'SECURE\_AUTH\_SALT', 'put your unique phrase here' );

define( 'LOGGED\_IN\_SALT', 'put your unique phrase here' );

define( 'NONCE\_SALT', 'put your unique phrase here' );

define('AUTH\_KEY', 'h5s9W]^]&ynFw=GQ?~UewHxR[~b!CRVItXh9y!XADkD%>@l^RHT~fVPBvAb-e-0W');

define('SECURE\_AUTH\_KEY', 'pO\_\_8G2 YYjPQ]==#ROUC,/$+1\_m|&|5rpe=i4DpX|q3gCbSSkE:2%TW\_+,uSd[]');

define('LOGGED\_IN\_KEY', 'ffbzMb&I;v5 t+i^Ce;+SD,C|MZfoD -T&Eh M/LPE+[`S-vd]9B.}UKSo5^VGkJ');

define('NONCE\_KEY', 'c;Y70y|-)1Qm^6[)-Ybn{(@Ro&n\_Tv!{.9vyzRB.Ix+2O(1}Th%~t -=|xiww5h}');

define('AUTH\_SALT', 'HQ(Y1:e-T|rt4MtU-R:l^!Y%:78$a/gF`/ytoEf;ZMU!@)O@7\*a>E\_E#|cwF0~kq');

define('SECURE\_AUTH\_SALT', 'qCykJlGR|+Oe}` y$3,f:.5FHiX\*+ |pl)vg~gy\_!i9^&=GLYIO+>JD\*n?a>|<WK');

define('LOGGED\_IN\_SALT', '%cr/[SYgM)MbMR$3z%ndIWJL6ugt>yPl| .mc1<lH$CE5;is7p-,Q}b.ZzmBH\*Dd');

define('NONCE\_SALT', '37:)iP\*F+Vti1.QWH-nT+0k+is\*!& yGtU>&oh)w?eH$4{M=LG9]T&2K9b{6YPO-');

### 

### Go to [this link](https://api.wordpress.org/secret-key/1.1/salt/) to generate values for this configuration section. You can replace the entire content in that section with the content from the link.

### https://api.wordpress.org/secret-key/1.1/salt/

### 

You can save and exit from nano by entering CTRL + O followed by CTRL + X.

With the configuration updated, you are almost ready to deploy your WordPress site. In the next step, you will make your WordPress site live.

### Deploying WordPress

### First, install the application dependencies you need for WordPress. In your terminal, run the following command.

### sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

### Second, change to the proper directory by running the following command:

### Cd /home/ec2-user

### Then, copy your WordPress application files into the /var/www/html directory used by Apache.

### 

### Finally, restart the Apache web server to pick up the changes.

### Sudo systemctl restart httpd

### 

### You should see the WordPress welcome page and the five-minute installation process.

### <https://aws.amazon.com/getting-started/hands-on/deploy-wordpress-with-amazon-rds/module-five/?refid=gs_card>

### Open public ip in browser

### 

### 

### Install wordpress

### 

### Installed successful

### 

### Login

### 

### Logged in successful

### 