RDS Exercise:

* Create a RDS (Mysql)
* In the already created EC2 instance, install apache2, mysql and php and restart apache.
* In the instance move the proper configuration files to the proper folders
* In AWS Console open the mysql port using the source of the security group
* In the instance configure wp-config.php to point the RDS
* Get the public IP of the server and access it via web browser
  + Configure WP, creating the username, password, etc, and hit install.
* If all is ok, you have to have access to wp configuration website.

**Set Up a WordPress Site Using EC2 and RDS**

**Introduction**

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.

**Solution**

Log in to the AWS Management Console using the credentials provided on the lab instructions page. Make sure you're using the *us-east-1* Region.

**Create RDS Database**

1. In the AWS management console, enter "rds" into the search bar on top.
2. From the results, select **RDS**.
3. Click **Create database**.
4. On the *Create database* page, set the following parameters:
   * Select **Standard create**.
   * Under *Engine options*, select **MySQL**.
   * Under *Templates*, select **Free Tier**.
   * Under *DB instance identifier*, enter "wordpress" and copy this into your clipboard.
   * Paste in "wordpress" as the **Master username** and **Master password**.
   * Under *VPC security group*, ensure Choose existing.
   * Under *Existing VPC security group*, select the **non-default security group** from the dropdown menu and remove the default security group.
   * Under *Availability zone*, select **us-east-1a**.
   * Expand *Additional configuration* and, under *Initial database name*, enter "wordpress".
   * Under *Backups*, uncheck the **Enable automatic backups option**.
   * Click **Create database**.
5. While the database is created, enter "ec2" in the search bar on top.
6. From the results, right-click **EC2** and open it in a new browser window or tab.
7. Under Resources, click **Instances (running)**.
8. Click the checkbox next to **webserver-01**.
9. In the top right, click **Connect**.
10. Click **Connect**.

**Install Apache and Dependencies**

1. In the terminal, install the Apache 2 web server, libraries, PHP, and PHP MySQL:
2. sudo apt install apache2 libapache2-mod-php php-mysql
3. When prompted, press **Y** for yes and press Enter.
4. Go into the newly created /var/www directory:
5. cd /var/www/
6. View the contents of the directory:
7. ls
8. Put wordpress into its own folder in the /var/www directory that we're currently in:
9. sudo mv /wordpress .
10. View the contents of the directory:
11. ls
12. Move into the wordpress directory:
13. cd wordpress
14. View the contents of the directory:
15. ls
16. Move the Apache configuration file into /etc/apache2/sites-enabled/ to enable the WordPress website to work from /var/www/wordpress:
17. sudo mv 000-default.conf /etc/apache2/sites-enabled/
18. Restart the Apache 2 configuration:
19. sudo apache2ctl restart

**Configure WordPress**

1. Open the WordPress config PHP file for editing:
2. sudo nano wp-config.php
3. Return to the browser window or tab that has the RDS Databases open.
4. Click the **wordpress** database.
5. In the *Connectivity & security* tab, under *Endpoint*, copy the endpoint provided into your clipboard.
6. Return to your terminal. Change the line define('DB\_HOST', 'localhost'); to read:
7. define('DB\_HOST', '<INSERT ENDPOINT HERE>');
8. Save and exit by pressing Control + X, followed by Y, and pressing Enter.

**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.
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.
9. Click **Save rules**.

**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*.
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:
   * *Site Title*: "A Cloud Guru"
   * *Username*: "guru"
   * *Password*: Select a strong password to use here, and make sure to copy it in your clipboard for later.
   * *Your Email*: "[test@test.com](mailto:test@test.com)"
5. Click **Install WordPress**.
6. Click **Log in**.
7. Enter "guru" for the *Username or email* and paste in the password that you copied earlier.
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.

**Conclusion**

Congratulations — you've completed this hands-on lab!