

Main Steps

1. Create aws account
2. Set up amazon aurora mysql cluster and ec2 instances
3. Connect local machine to ec2 instance using ssh
4. Connect to the Aurora mysql database
5. Test connection

Create aws account

This step should be trivial. No comments provided.

Set up amazon aurora mysql cluster and ec2 instances

1. Navigate to the RDS console
 - a. In the AWS Management Console, use the search bar to find “RDS” and select it.
2. Create an ec2 instance
 - a. Navigate to EC2 console
 - i. In the AWS Management Console, use the search bar to find “EC2” and select it.
 - b. Launch ec2 instance
 - i. Application and OS Images
 1. Quick start - amazon linux
 2. Amazon machine image - amazon linux 2023 AMI
 3. Architecture - 64-bit (x86)
 - ii. “Instance type”
 1. Select “t2.micro”
 - iii. “Key pair (login)” - create new key pair
 1. Key pair type - select “RSA”
 2. “Private key file format” - select “.pem”
 3. There will be a file that is downloaded on your computer. Store this in a safe and accessible location.

- c. Network settings
 - i. Firewall (security groups)
 - 1. Select “create security group”
 - 2. Toggle ssh traffic from “anywhere 0.0.0.0/0”
 - d. Configure storage
 - i. I picked “8” and “gp3”. Decide based on what you need.
 - e. Advanced detail
 - i. IAM instance profile - select the ssm access role you made earlier
3. Create a new database
- a. Click on the “Create database” button.
 - b. Select the following options:
 - i. Database creation method
 - 1. Standard create
 - ii. Engine options
 - 1. Engine type - Aurora (mysql compatible)
 - 2. Engine version - Aurora mysql 3.05.2 (compatible with mysql 8.0.32) - default for major version 8.0
 - 3. Templates - production
 - iii. Settings
 - 1. Credential settings - just toggle “self managed” under credentials management and make a password
 - iv. Cluster storage configuration - I used “aurora standard”, but if spectrum needs a very large computing power, feel free to use “aurora i/o optimized”
 - v. Instance configuration - toggle serverless
 - vi. Availability & durability - toggle “create an aurora replica or reader node in a different AZ”
 - vii. Connectivity
 - 1. Compute resource - select “connect to an ec2 compute resource” and pick the ec2 instance you made earlier

2. Network type - Either "IPv4" or "dual stack mode" should work, but i used the former
- c. Now create your database!
4. Connect your ec2 instance to your database
 - a. Go to aws management console and search up "rds"
 - b. Select "db instances"
 - c. Click on the db identifier of the most general db
 - d. Scroll to "connected compute resources"
 - e. Click actions and "set up ec2 connection"
 - f. Choose your ec2 instance you created and click continue

Connect local machine to ec2 instance using ssh

Go into terminal and type this command:

ssh -i "your-key-file.pem" ec2-user@your-ec2-public-ip

- Replace "your-key-file.pem" with the path to your private key file (this is the file you saved during the key pair login step) and "your-ec2-public-ip" with the EC2 instance's public IP address and "ec2-user" with the username associated with your ec2 instance (default username is ec2-user)
- Your ec2's public ip address can be found through the following process
 1. Go to aws management console, search up ec2 to get to your ec2 dashboard, and select "instances"
 2. Select the id of the instance you're working with
 3. Copy the code under "public IPv4" address

Connect to the Aurora mysql database

In your ec2 instance terminal, enter the following command:

```
mysql -h [writer-endpoint] -u [MasterUsername] -p
```

- Replace “[writer endpoint]” with the endpoint of your writer instance of your db and replace “[master username]” with the username of your db.
- To find the endpoint of your writer instance of your db,
 - Navigate to your databases and click the db identifier that says “writer instance” beside it.
 - Switch to “connectivity and security” tab
 - Copy the endpoint
- To find the username of your db, click on the most general db, go to configuration, and copy the thing under “master username”

Test connection

Congratulations on getting through this module!

Now you can test your database by just entering some sql commands.