

AWS Deployment

Creating a RDS instance

1. Go to AWS Console. Search for Aurora and RDS.
2. Then click on create database.
3. Use Standard create.
4. Select MySQL
5. **Settings** > **DB instance identifier** : It will be the name of database instance. You can change it anything you want.
6. **Settings** > **Credentials Settings** > **Master username** It will be user name for accessing the database. By default it is set to **admin**. You can change it anything.
7. **Settings** > **Credentials Settings** > **Master password** It will be the password for accessing the database. Create your password.
8. **Connectivity** > **Public access** . Set it to **Yes**.
9. **Connectivity** > **VPC security group (firewall)**. Click on **Create new**. Then **New VPC security group name** . Just add a name for security group.
10. Then click on **create database**. It will take some time to create.
11. When its status changes to **Available**. Click on the instance. Then **Connectivity & security** > **Endpoint & port** > **Endpoint**. Copy that endpoint.
12. Now we will test whether everything is working or not. So first make sure **mysql** is present on your device. Then open cmd

and run this command

```
mysql -h <Endpoint> -p 3306 -u <master-username> -p
```

Then hit the enter

It will ask you to enter password

Enter the **Master Password** that you have created earlier. If all goes well it will connect you to rds instance.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 29
```

Server version: 8.0.43 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

This type of message is shown. It is just an instance you have to create database so that you can store data.

Creating EC2 instance

1. Go to AWS Console. Search for EC2.
2. Then click on **Launch an instance**.
3. Then under **Name and tags > Name**. This will be name of your ec2 instance. Add a name for your ec2 instance.
4. Then **Key pair (login) > Key pair name > Create new key pair**. It will create a pop-up window **Create key pair > Key pair name**. Enter a name for the key. After that click on **Create Key Pair**. It will download a file with **.pem**. It will be the key that will help us to connect to ec2 instance.
5. Then **Network settings > Firewall (security groups)**. Make sure to Select the **Allow SSH traffic from** and **Allow HTTP traffic from the internet**.
6. Now click on **Launch instance**. It will start creating your instance.
7. Now when **Instance state** is **Running**. It means your ec2 instance has been setup and it is working.
8. Now we will check that whether we can connect to the instance or not to make sure everything is fine. First click on **Instance Id** of the instance we created in the **Instances**. Then click on **connect > SSH Client**. Remain this window open.
9. Now open a new **cmd**. Now change your directory to place where your key that you have downloaded is stored. Then again go back to the window. Copy the command in **Step 3**. Paste in **cmd** and then run.
10. Go to back to the window again and copy the command in the example and paste in the cmd and run. Enter **yes**. If all goes well it will connect you to ec2.

Packaging spring boot app

1. Open a new terminal Change directory to the your spring boot app.
2. Enter command **./mvnw clean package**. It will create a **.jar** file for your project inside the target jar file. Now we need to upload the **projectName.jar** in the ec2 instance. So We in the terminal where we

have written this packaging command. Enter this

```
scp -i your-key.pem <jar-file> ec2-user@<ec2-public-dns>:/home/ec2-user/
```.
Since you are at root your .jar file location will be `target/projectName.jar`.
EC2 Public DNS is available in `PUBLIC DNS` in the instance details of the window.
It will copy the .jar file into your ec2 Instance.
```

```
3. We can make sure this by going to the terminal in which we connected to
ec2 instance and then running `ls` command it will show the jar file.
```

```
4. Now we need to first install java to run application. Run these command
inside the terminal which is connected to ec2 instance.
```

```
```bash
sudo yum update -y
sudo yum install java-17-amazon-corretto -y
java --version
```

5. Now we need to add application.properties file inside ec2 instance. So run this command.

```
nano application.properties
```

6. Now paste this content.

```
spring.datasource.url=jdbc:mysql://<rds-database-Endpoint>?
useSSL=false&allowPublicKeyRetrieval=true
spring.datasource.username=<Master-username>
spring.datasource.password=master-password

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
server.port=3000
```

Replace the `<rds-database-Endpoint>`, `<Master-username>` and `master-password` with the required values.

7. Now use `ctrl+x > y` > Hit Enter . It will create application.properties file.

8. Now we first need to add the security group of the ec2 to rds so that they can connect

```
Go to EC2 → Instances → Your EC2 → Security → Security group.
```

Note the EC2 security group ID (e.g., sg-xxxxxx).

```
Go to RDS → Security Groups → dbfirewall → Inbound rules → Edit inbound rules.
```

Add a rule:

Type	Protocol	Port Range	Source
MySQL/Aurora	TCP	3306	EC2 security group ID

4. Now go back to terminal where we have connected to ec2 instance.

```
java -jar your-jarfile.jar --server.address=0.0.0.0 --server.port=3000 --spring.config.location=application.properties
```

5. Now we need to expose this port 3000 to users so that they can access this application

```
Go to EC2 → Instances → Your EC2 → Security → Security group.
```

Edit Inbound rules:

Type	Protocol	Port Range	Source
Custom TCP	TCP	3000	0.0.0.0/0

Save it.

Now go to EC2 > Instances > your instance id > Public IPv4 address . Copy Public IPv4 address.

Now `http://<Public IPv4 address>:3000/`

This will allow you to access your app. Now you can test your app in the postman.