Assignment 5 – RDS

Task1. Create an Aurora Database and connect to it from an EC2

1. Create database SG – Allow inbound access from the EC2
   a. Go to EC2 -> In the sidebar, there are Security groups, click on that -> Click Create Security Group
   b. Give name as MyRdsSg -> give description -> select your VPC
   c. In the inbound rules section, click on add rule
   d. Click on Type -> Look for MYSQL/Aurora
   e. Source -> select the SG of the web server that accesses the database -> Create Security Group
2. Create RDS database
   a. Templates -> select dev/test
   b. Enter the master password. That you will use when logging into the DB. The username is admin.
   c. DB instance class -> Select Burstable classes -> Select db.t3.small
   d. Availability & durability -> Create an Aurora Replica
   e. Select Create an Aurora Replica or Reader node in a different AZ in Availability & Durability section.
   f. Select your VPC, it will automatically select the subnet. Select SG as well.
   g. you must uncheck Enable Enhanced monitoring in the Monitoring panel
3. Install mysql client on EC2

```
sudo yum install mysql -y
```

4. Connect to the RDS instance.

```
mysql -h <endpoint_url> -P 3306 -u root -p
```

5. Create a table and insert some records.

```
show databases;
create database university;

CREATE TABLE TEACHER (
    TEACHER_ID int,
    NAME varchar(255)
);

CREATE TABLE COURSE (
    COURSE_ID int,
    COURSE_CODE varchar(255),
    COURSE_NAME varchar(255),
      TEACHER_ID int
);
```

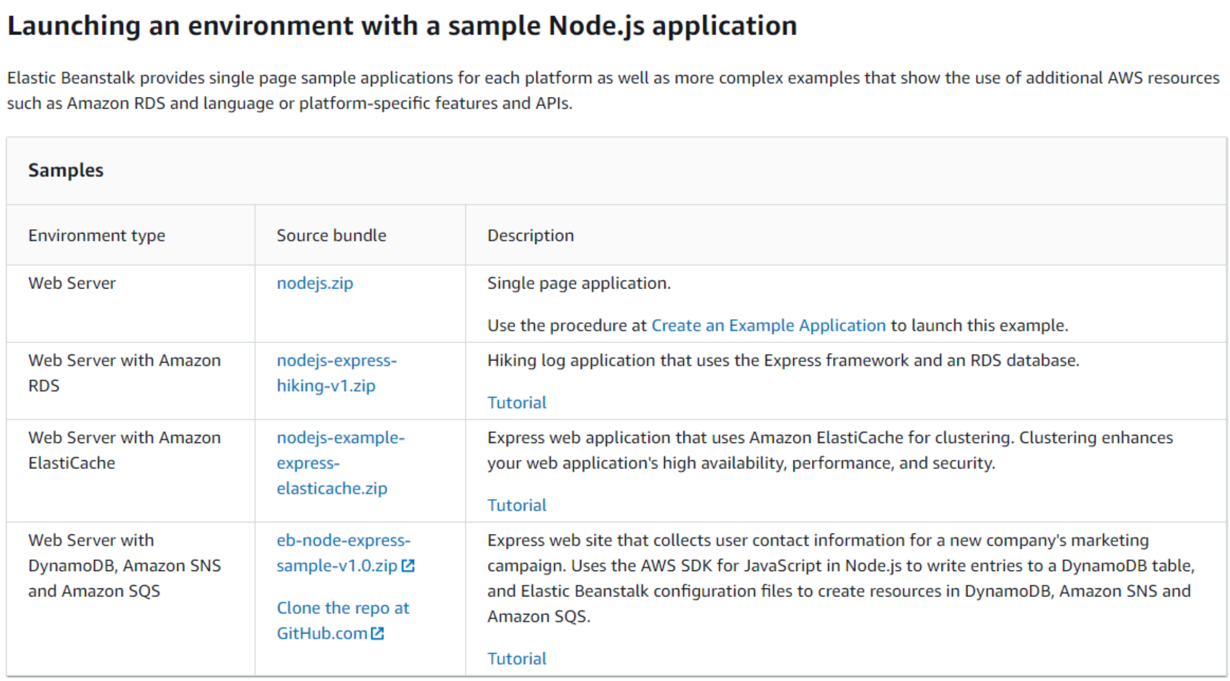
```
INSERT INTO TEACHER (TEACHER_ID, NAME)
VALUES (1, "UNUBOLD"),
(2, "ASAAD"),
(3, "UMUR");

INSERT INTO COURSE (COURSE_ID, COURSE_CODE, COURSE_NAME, TEACHER_ID)
VALUES (1, "CS516", "CLOUD COMPUTING", 1),
(2, "CS568", "React", 1),
(3, "CS569", "Angular", 1),
(3, "CS569", "Angular", 2);

SELECT TEACHER.NAME, COURSE.COURSE_ID, COURSE.COURSE_NAME
FROM TEACHER JOIN COURSE ON TEACHER.TEACHER_ID = COURSE.TEACHER_ID;
```

Refer: Connecting to a DB instance running the MySQL database engine

6. Connect to the read instance and run some queries.

Task 2. Run NodeJS and MySQL app on Elastic BeanStalk

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/nodejs-getstarted.html



Launching an environment with a sample Node.js application

Elastic Beanstalk provides single page sample applications for each platform as well as more complex examples that show the use of additional AWS resources such as Amazon RDS and language or platform-specific features and APIs.

Samples

Environment type	Source bundle	Description
Web Server	nodejs.zip	Single page application. Use the procedure at Create an Example Application to launch this example.
Web Server with Amazon RDS	nodejs-express-hiking-v1.zip	Hiking log application that uses the Express framework and an RDS database. Tutorial
Web Server with Amazon ElastiCache	nodejs-example-express-elasticache.zip	Express web application that uses Amazon ElastiCache for clustering. Clustering enhances your web application's high availability, performance, and security. Tutorial
Web Server with DynamoDB, Amazon SNS and Amazon SQS	eb-node-express-sample-v1.0.zip Clone the repo at GitHub.com	Express web site that collects user contact information for a new company's marketing campaign. Uses the AWS SDK for JavaScript in Node.js to write entries to a DynamoDB table, and Elastic Beanstalk configuration files to create resources in DynamoDB, Amazon SNS and Amazon SQS. Tutorial

[Extra] ECS and PhpMyAdmin

- Spin up PhpMyAdmin in your AWS account then use PhpMyAdmin to connect to your DB instead of a Bastion server.
    - Change the configuration in the image. Then it will pick up the endpoint and there will be 3 fields, URL, Username, and Password.

- Write a docker image from https://hub.docker.com/r/phpmyadmin/phpmyadmin/
- Upload the image to AWS ECR
- Deploy this on ECS Fargate
- Check SG and IAM configs