# CSE 5234 – Distributed Enterprise Computing

Lab 12 – AWS Cloud databases & Infrastructure as Code

The goal of this lab is to give students a working knowledge of databases in AWS, and infrastructure as code.

Objectives

1. Prerequisite - CREATE A FREE AWS ACCOUNT
2. Setting up CloudFormation relational data store

Cloudformation RDS Lab

**Prerequisites:**

* **A free AWS account. Please follow lab 0 instructions to create one.**
* **Download the rds\_cfn.yaml file as a starting point for this lab.**

1. Log into the AWS Console, you should see a screen like this

A screenshot of a cell phone

Description automatically generated

1. Click on “Services” in the upper right, search for Cloudformation, and select it.

Text

Description automatically generated

1. Select “Create Stack” on the Cloudformation console.

Graphical user interface, text

Description automatically generated

1. Select “Upload a Template File” and then choose your rds\_cfn.yaml file.

Graphical user interface

Description automatically generated

1. Name your stack and select next on this page and the next, before clicking submit at the bottom of the review screen.

A screenshot of a computer screen

Description automatically generated

1. Watch the stack deployment in the console. This deployment **will fail**. Once the stack hits the stage “Rollback Complete”, delete the stack.

Graphical user interface, text

Description automatically generated

1. Using the documentation [here](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-rds-database-instance.html) fill out the required properties of the instance in your cloudformation. To ensure that you stay within the free tier, use “postgres” as your Engine, “db.t2.micro” as your DbInstanceClass, and “5” for your Allocated storage.

Your code should look something like this:

Properties:  
 DBName: "osutest"  
 Engine: "postgres"  
 MasterUsername: "pocsad2"  
 DBInstanceClass: "db.t2.micro"  
 AllocatedStorage: "5"  
 MasterUserPassword: "TestPass\_123"

Once you have this, repeat the steps to create a Cloudformation stack from above. It will take 5-10 minutes to complete

1. This stack should deploy successfully, lets check the resources. Click “Resources” on your Cloudformation stack and then click the “Physical ID” for your database.

A screenshot of a computer screen

Description automatically generated

This will take you to the RDS console where you can see the attributes of your database. Take a moment to explore the console.

A screenshot of a computer screen

Description automatically generated

1. Now that our database is live, we can improve the Cloudformation for our instance. First let’s start by parameterizing the properties of out database. Documentation on Cloudformation parameters can be found [here](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html). We can parameterize each input to our database. Once you add parameters you code should look like this.

Parameters:  
 pDatabaseName:  
 Type: String  
 Description: Name of Database  
 pEngine:  
 Type: String  
 Description: Engine for database  
 AllowedValues:  
 - mysql  
 - postgres  
 pMasterUserName:  
 Type: String  
 Description: Master Username  
 pMasterUserPassword:  
 Type: String  
 Description: Master User Password  
 NoEcho: True  
 pDbClass:  
 Type: String  
 Description: Database Instance Class  
 AllowedValues:  
 - db.t2.micro  
 pStorageSize:  
 Type: String  
 Description: Allocated Storage Size  
 Default: '5'  
  
Resources:  
 rDBInstance:  
 Type: AWS::RDS::DBInstance  
 Properties:  
 DBName: !Ref pDatabaseName  
 Engine: !Ref pEngine  
 MasterUsername: !Ref pMasterUserName  
 DBInstanceClass: !Ref pDbClass  
 AllocatedStorage: !Ref pStorageSize  
 MasterUserPassword: !Ref pMasterUserPassword

Now that you have updated your Cloudformation file we can update the stack in console.

A screenshot of a computer screen

Description automatically generated

Once here click “Replace current template” and then reupload your yaml file.

Graphical user interface, application

Description automatically generated

Click next and you will be prompted for input to all the parameters you defined. Fill them in appropriately. Continue through the process as before.

A screenshot of a computer screen

Description automatically generated

Once the stack has updated you can check your resource again, in the case of RDS, the update action creates a new resource, so there will be a new database to go to, with all the same attributes.

The benefits of parameterizing your stack input is being able to bypass uploading a new template to the stack for each change, as you can select “Use existing template” on updates and only change the parameters. This allows for quick and easy changes of things like storage size or instance class while also providing a log of how these changes were made and when.

1. Now that we have updated the stack with parameters, we can start to add other resources that will improve our infrastructure. We will be adding 4 things to our Cloudformation stack: AWS Secrets Manager secret, an EC2 Security Group, an RDS DB Security Group and Cloudformation Outputs. Documentation for each of these can be found below:  
   A. <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-secretsmanager-secret.html>

B. <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-security-group.html>

C. <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-rds-security-group.html>

D. <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/outputs-section-structure.html>

Once you have read through this you will want to add a secret that contains your master user and password so that you can systematically retrieve this information incase it is ever lost. Your code should look similar to this when done:

rMasterUserSecret:  
 Type: AWS::SecretsManager::Secret  
 Properties:  
 Name: RdsMasterUserInfo  
 Description: This secret has a hardcoded password in SecretString (use GenerateSecretString instead)  
 SecretString: !Sub '{"username":${pMasterUserName},"password":${pMasterUserPassword}}'

The string for SecretString should be on a single line,

You will also want to create an EC2 security group that allows front end access to your database. That code should look like this:

rWebServerSecurityGroup:  
 Type: AWS::EC2::SecurityGroup  
 Properties:  
 GroupDescription: Enable HTTP access via port 80 and SSH access  
 SecurityGroupIngress:  
 - IpProtocol: tcp  
 FromPort: '80'  
 ToPort: '80'  
 CidrIp: 0.0.0.0/0  
 - IpProtocol: tcp  
 FromPort: '22'  
 ToPort: '22'  
 CidrIp: 0.0.0.0/0

Once you have this you can create your RDS DB security group, it should look like this:

rDBSecurityGroup:  
 Type: AWS::RDS::DBSecurityGroup  
 Properties:  
 DBSecurityGroupIngress:  
 - EC2SecurityGroupName: !Ref rWebServerSecurityGroup  
 GroupDescription: Frontend Access

Now you will need to update your code for the instance itself to use the DB Security Group, it should look like this now:

rDBInstance:  
 Type: AWS::RDS::DBInstance  
 Properties:  
 DBName: !Ref pDatabaseName  
 Engine: !Ref pEngine  
 MasterUsername: !Ref pMasterUserName  
 DBInstanceClass:  
 Ref: pDbClass  
 DBSecurityGroups:  
 - !Ref rDBSecurityGroup  
 AllocatedStorage: !Ref pStorageSize  
 MasterUserPassword: !Ref pMasterUserPassword  
 StorageEncrypted: True

Once all of this is done, at the bottom of your file you will want to add a Cloudformation output for the database id and the secret id so that you can reference them. That should look like this:

Outputs:  
 oInstanceId:  
 Description: InstanceId of the newly created RDS Instance  
 Value: !Ref rDBInstance  
 oSecretId:  
 Description: Secret id of master user secret  
 Value: !Ref rMasterUserSecret

Now it is time to update the stack, follow the same instructions from the last update to update with the new code. When your stack has completed its upload take a look at the resources tab again to see all of your new resources.

**Epilogue**: Now you have all your necessary resources deployed! From here you can follow tutorials from Amazon on how to interact with your RDS instance, and how to optimize it for performance and cost. When you are done with your infrastructure remember to delete your Cloudformation stack to avoid incurring any costs after your AWS Free Tier period ends.