

Lab Brief

Course: Managed services on AWS

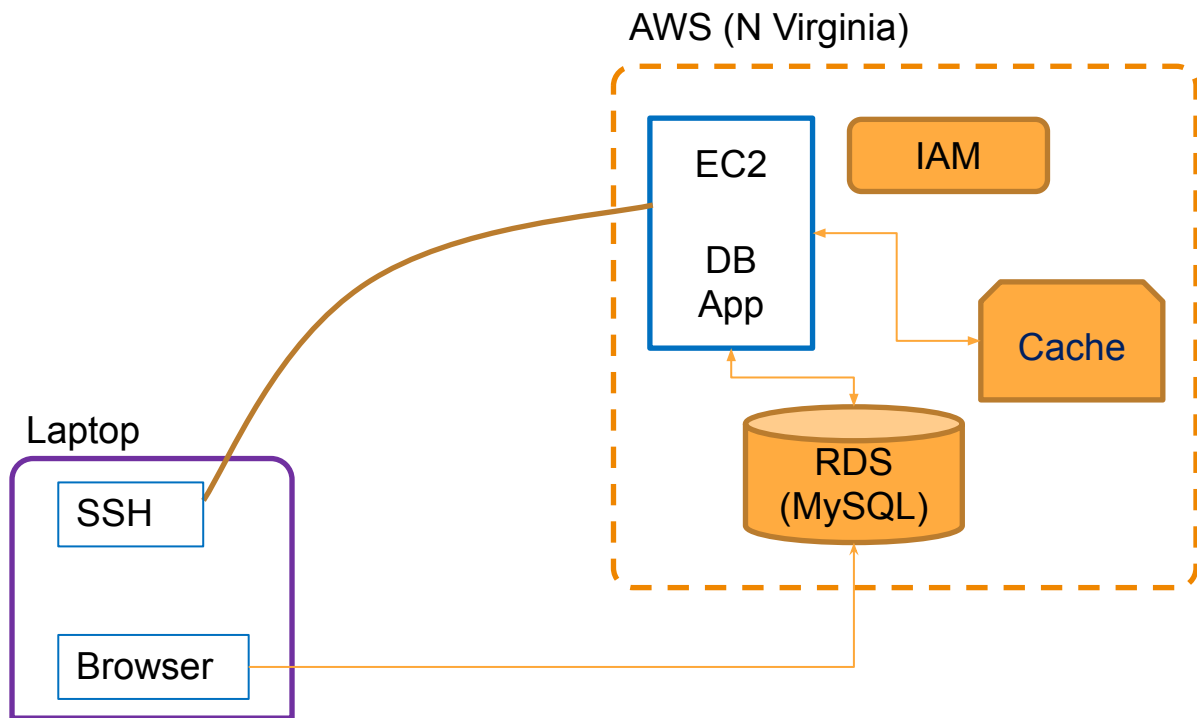
RDS | EC2 database program

(Create a MySQL instance using RDS and access it using a custom program from an EC2 instance using an appropriate role)

Learning Outcomes

1. Get an EC2 instance using the 7 step workflow & SSH in it
2. Create a MySQL database using RDS
3. Write the custom application and deploy on EC2
4. Able to apply roles to EC2 instances

Final Goal



What is needed?

1. AWS Account Credentials
2. EC2 Instances (Linux)
3. Shell script environment (any text editor of your choice)
4. Full access to - EC2, RDS, IAM, ElastiCache

How to do it? - 1

1. Ensure your region is set to "N Virginia"
2. Create a private MySQL RDS
 - a) Create a "Dev/Test" MySQL instance
 - b) Select "Non publicly" available DB
 - c) DB instance = "flipbasket", UID/PWD = root/password
 - d) AZ = 2a, create a new SG
 - e) **IMP** When the RDS instance is created check the SG to ensure that the "inbound" rule does not have any specific IP address, instead use the VPC CIDR block of the EC2 instance. Otherwise we will not be able to connect from the EC2 instance
3. Create 1 EC2 instance using the 7 step workflow
 - a) Use the usual Ubuntu 20.04 LTS AMI in AZ1
 - b) Download a new PEM file and SSH to the instance
4. IAM
 - a) "Choose the service that will use this role" = EC2
 - b) "Select use case" = "EC2" (the 1st option)
 - c) Select the policy "Amazon RDS full access"
 - d) Create the role "ec2-multirole"
 - e) Assign to the EC2 instance

This file is meant for personal use by demay.tom@gmail.com only.

Sharing or publishing the contents in part or full is liable for legal action.

How to do it? - 2

1. SSH to the EC2 instance and fire the following commands
 - a) `sudo apt update`
 - b) `sudo apt install mysql-client`

2. Connect to the RDS instance & create the data
 - a) `wget https://d6opu47qoi4ee.cloudfront.net/employees.sql`
 - b) `mysql -h [RDS end point] -u admin -p`
You should get the MySQL prompt after entering the password
 - c) `create database employees;`
 - d) `use employees;`
 - e) `source employees.sql`
 - f) `describe employees;`
 - g) Note the column names in a text editor from the output of the above command
 - h) Close the mysql terminal

How to do it? - 3

- Follow the steps for the Python program connecting to the RDS instance from EC2
 - `sudo apt install python2-minimal`
 - `sudo apt install python3-pip`
 - `sudo pip3 install boto3`
 - `sudo pip3 install mysql-connector-python`
 - `sudo pip3 install pymysql`
- Use the `rds.py` as the template and update the hostname to the endpoint of your RDS instance and run the program
 - `wget https://d6opu47qoi4ee.cloudfront.net/rds.py`
 - Modify the python program in all the places marked with "TBD"
 - `python3 rds.py`
 - In case you get an error for the mysql connector just re-execute the `sudo pip install` command
 - **Q: state your observations when you remove the role and rerun the program (Hint - RDS SG)**

In case the pip install fails execute the following 3 lines -
`export LC_ALL="en_US.UTF-8"`
`export LC_CTYPE="en_US.UTF-8"`
`sudo dpkg-reconfigure locales`

How to do it? – 4 (Advanced *)

- Once you have updated the program to connect with the database, do the following
 - Create a Redis cluster (self managed / Elasticache)
 - Modify the program to include another method "getOrder(...)"
 - Add a class that represents the order
 - This method needs to check if the order exists in the cache, if found then get from cache and return else go to the DB, put in cache and return
 - Use the python modules - redis & pickle

* This step of the assignment is NOT mandatory, and will not have any impact on the grades. You can choose to do / skip it only from the learning perspective.

Mandatory step: Resource Clean-Up

1. Cloud is always **pay per use model** and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
2. **After completing with the lab, make sure to delete each resource created in the reverse chronological order.**
3. Check resources in each cloud region that you have worked on before logging off.
4. Since the dashboard doesn't show cross-region resources, it is up to you to find and delete them.