Names: Mohamad Laham, ID:20677433, Moslem Masarwa, ID: 214023798

XYZ University Lab Project – AWS Environment Setup Guide

This document provides detailed, step-by-step instructions for setting up your AWS infrastructure. Complete each section in sequence and verify each setup before proceeding.

1. VPC and Networking Setup

• Create VPC

- CIDR block: 10.0.0.0/16

• Subnets

- Public Subnet 1: 10.0.0.0/27 in us-east-1a
- Public Subnet 2: 10.0.0.32/27 in us-east-1b
- Private Subnet 1: 10.0.0.64/27 in us-east-1a
- Private Subnet 2: 10.0.0.96/27 in us-east-1b

• Internet Gateway

- Create and attach to the VPC

• Route Table

- Create a public route table
- Add a route to 0.0.0.0/0 via the Internet Gateway
- Associate with the public subnets

2. RDS Database Setup

• Engine: MySQL

• Instance Type: db.t4g.micro

• DB Name: STUDENTS

• Username: nodeapp

• Password: student12

• Subnet Group: Select both private subnets

• VPC Security Group

- Inbound Rule: Allow MySQL (port 3306) only from EC2 security group

3. Secrets Manager

Store database credentials securely:

```
aws secretsmanager create-secret \
--name Mydbsecret \
--description "Database secret for web app" \
--secret-string '{ "user":"nodeapp", "password":"student12", "host":"<RDS-ENDPOINT>", "db":"STUDENTS" }'
```

4. EC2 Web Server Setup

- AMI: Ubuntu 24.04 LTS
- Instance Type: t2.micro
- IAM Role: Attach LabRole
- Security Group: Allow inbound HTTP (port 80) from anywhere

User Data Script:

```
#!/bin/bash -xe
apt update -y
apt install nodejs unzip wget npm mysql-client -y
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCAP1-1-91571/1-lab-capstone-project-1/code.zip -P /home/ubuntu
cd /home/ubuntu
unzip code.zip -x "resources/codebase_partner/node_modules/*"
cd resources/codebase_partner
npm install aws aws-sdk
export APP_PORT=80
npm start &
```

5. Data Migration

```
mysqldump -h <EC2-DB-IP> -u nodeapp -p --databases STUDENTS > data.sql
mysql -h <RDS-ENDPOINT> -u nodeapp -p < data.sql
```

6. Load Balancer & Auto Scaling

• Application Load Balancer

  - Type: Internet-facing

  - Listener: HTTP on port 80

  - Target Group: Register EC2 instances

• Launch Template

  - Use a custom AMI created from your configured EC2

• Auto Scaling Group

  - Min: 2, Max: 4

  - Policy: Target tracking on CPU (e.g., 50%)

7. Load Testing (Validation)

Use Cloud9 or any EC2 with Node.js and NPM installed:

```
npm install -g loadtest
loadtest --rps 1000 -c 500 -k http://<Your_LOAD_BALANCER_DNS>
```

Final Notes

• All resources should be in the us-east-1 region

• Regularly monitor billing via the AWS Console

• IMPORTANT: Terminate all EC2 and RDS resources after testing to avoid ongoing charges