Name:AbdelRahman Moustafa Mohamed Abdallah

Group name: AWS Cloud Specialist (ALX1_SWD8_M1e)

Student ID: 1110242719

AWS Academy Lab Project - Cloud Web Application Builder

Building a Highly Available, Scalable Web Application

Supervise by: Eng. Merhan Adel

Badge Link

About Project

Deploy a web app on EC2, migrate the database to RDS, and ensure high availability and scalability using a load balancer and an Auto Scaling group.

Steps for the Project:

Phase 1: Planning the Design and Estimating Cost

1. **Creating an Architectural Diagram:**
   - Design an architecture diagram illustrating the AWS services and their interactions.
2. **Developing a Cost Estimate:**
   - Use the AWS Pricing Calculator to estimate the cost of running the solution in the us-east-1 region for 12 months.

Phase 2: Creating a Basic Functional Web Application

1. **Creating a Virtual Network:**
   - Set up a virtual private cloud (VPC) and necessary subnets for hosting the application.
2. **Creating a Virtual Machine:**
   - Deploy a virtual machine (EC2) with the latest Ubuntu AMI to host the web application.
3. **Testing the Deployment:**
   - Ensure the web application is accessible from the internet and functions correctly (view, add, delete, modify records).

Phase 3: Decoupling the Application Components

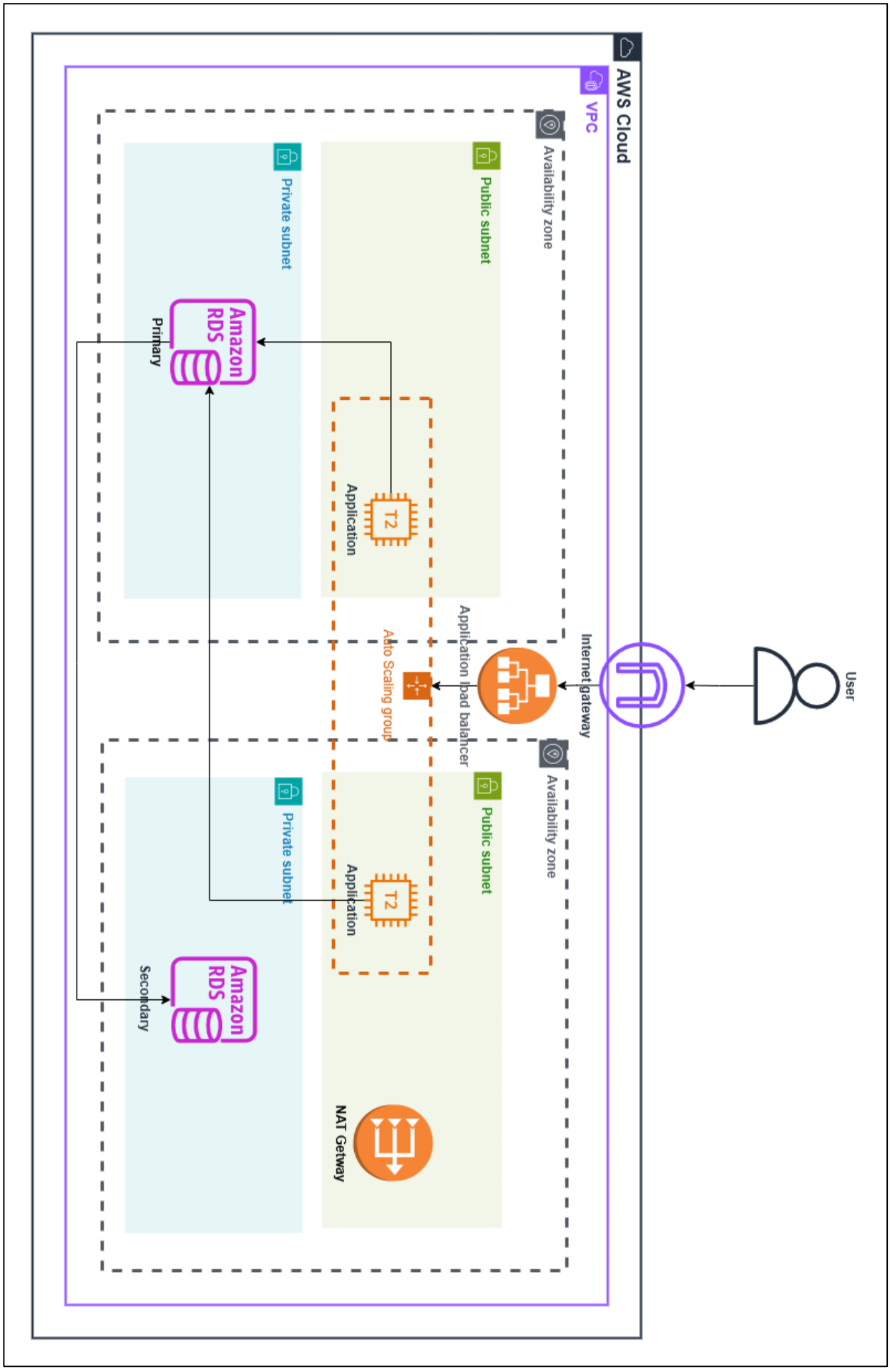
1. **Changing the VPC Configuration:**
   - Update or recreate the VPC to support separate hosting for the database and web server.
2. **Creating and Configuring the Amazon RDS Database:**
   - Set up an Amazon RDS instance running a MySQL engine, ensuring only the web application can access it.
3. **Configuring the Development Environment:**
   - Provision an AWS Cloud9 environment for executing AWS CLI commands.
4. **Provisioning Secrets Manager:**
   - Use AWS Secrets Manager to store database credentials securely and configure the web application to use these credentials.
5. **Provisioning a New Instance for the Web Server:**
   - Deploy a new EC2 instance for the web application or reconfigure the existing instance to connect to Amazon RDS.
6. **Migrating the Database:**
   - Migrate data from the original database on the EC2 instance to the new Amazon RDS database.
7. **Testing the Application:**
   - Verify the application functionality by performing CRUD operations on student records.

Phase 4: Implementing High Availability and Scalability

1. **Creating an Application Load Balancer:**
   - Set up an Application Load Balancer across at least two Availability Zones.
2. **Implementing Amazon EC2 Auto Scaling:**
   - Create a launch template and an Auto Scaling group to manage EC2 instances hosting the web application.
3. **Accessing the Application:**
   - Test application functionality by performing CRUD operations again.
4. **Load Testing the Application:**
   - Conduct a load test using scripts to simulate 1,000 requests to evaluate performance under load.

Phase 1: Planning the design and estimating cost

Task 1: Creating an architectural diagram Link Diagram



Task 2: Developing a cost estimate Link CSV File

1. **Application Load Balancer (ALB) :**
   - **Monthly Cost: $16.46**
   - **Annual Cost: $197.52**
   - **Quantity: 1**
2. **Amazon EC2 :**
   - **Monthly Cost: $40.16**
   - **Annual Cost: $481.92**
   - **Configuration: 4 t2.micro instances running on Linux, utilizing 700 hours per month, with 20 GB of EBS storage.**
3. **Amazon RDS for MySQL :**
   - **Monthly Cost: $14.20**
   - **Annual Cost: $170.40**
   - **Configuration: 1 db.t3.micro instance with 20 GB of gp3 SSD storage, deployed in a single availability zone.**

The Total Cost:

Monthly Payments:

- **Total Monthly Cost: $70.82 USD**
  You will pay $70.82 each month for the services provided (Application Load Balancer, EC2 instances, and RDS for MySQL).

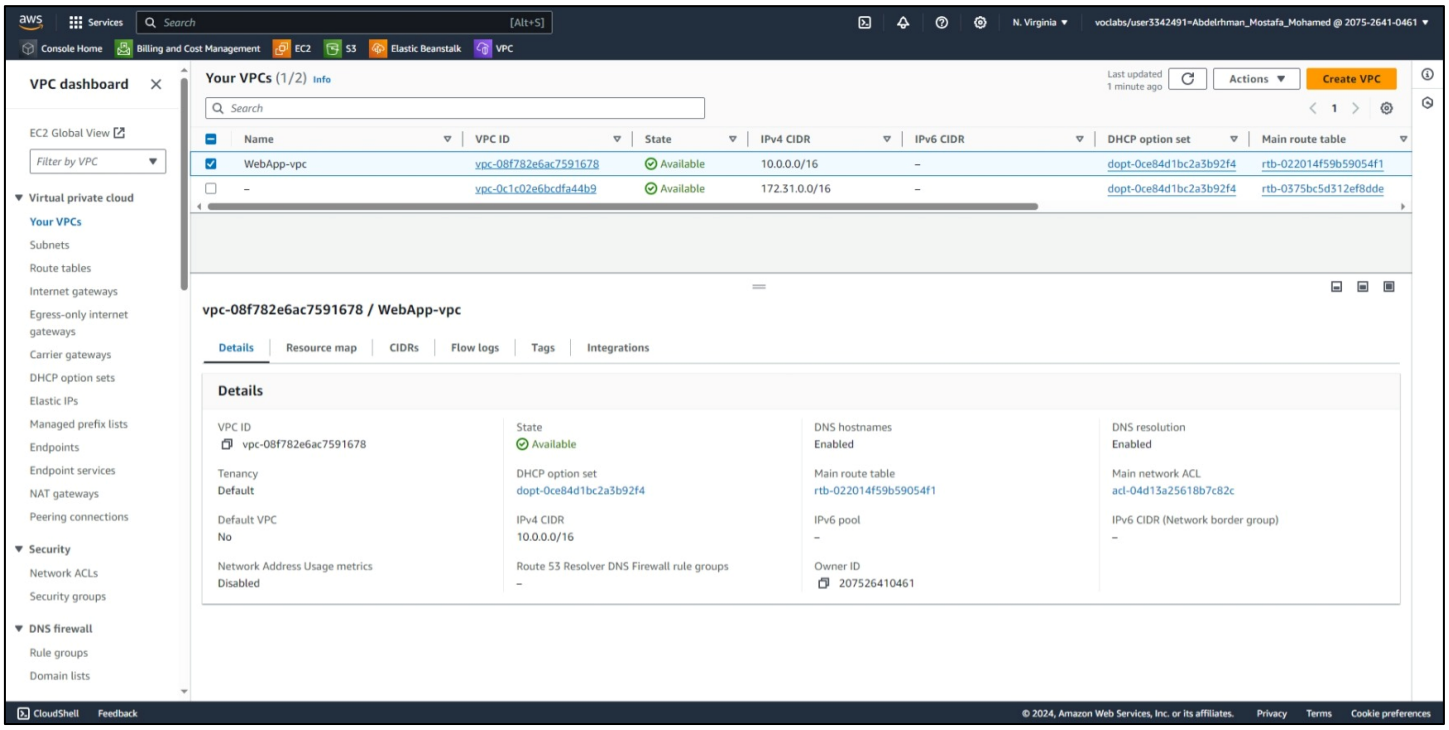
Yearly Payments:

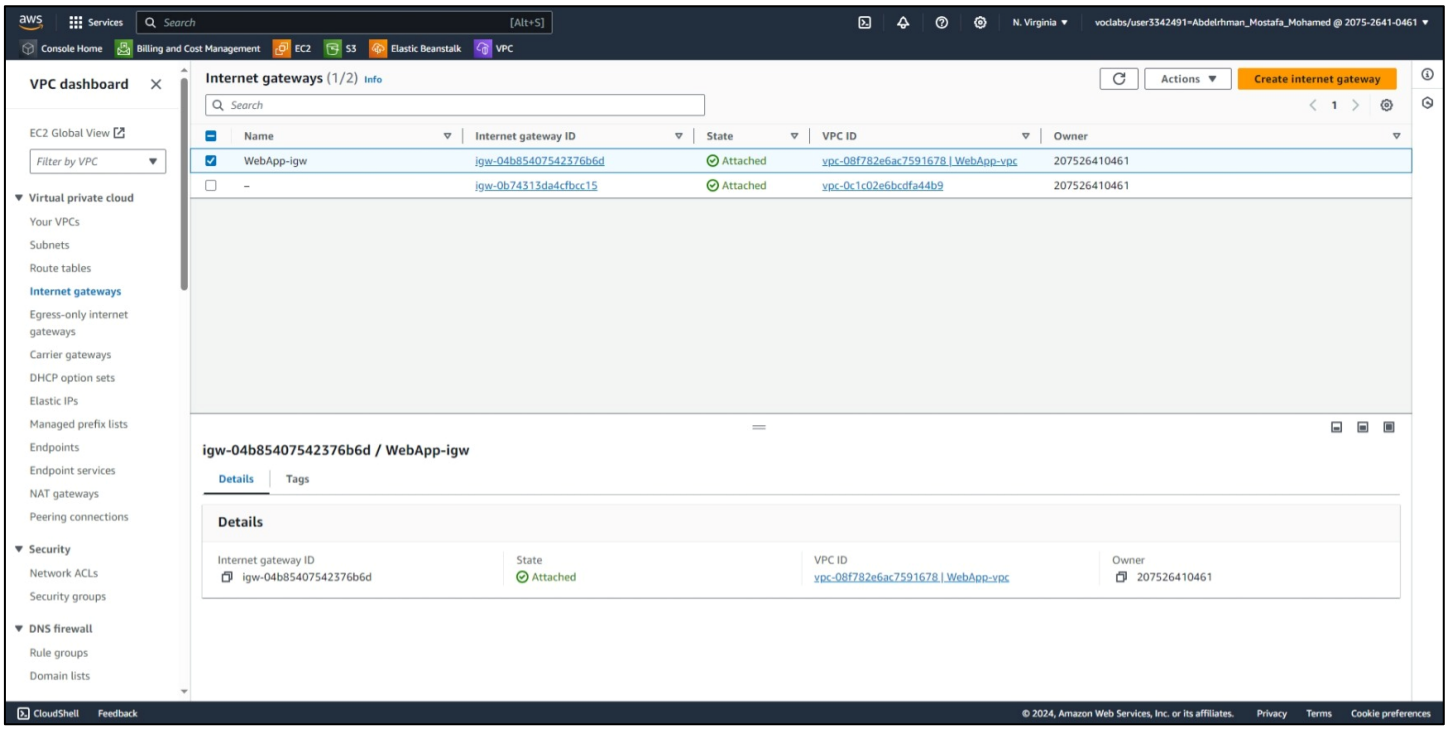
- **Total Yearly Cost: $849.84 USD**
  If you choose to pay annually, the total amount due for the 12 months will be $849.84 USD.

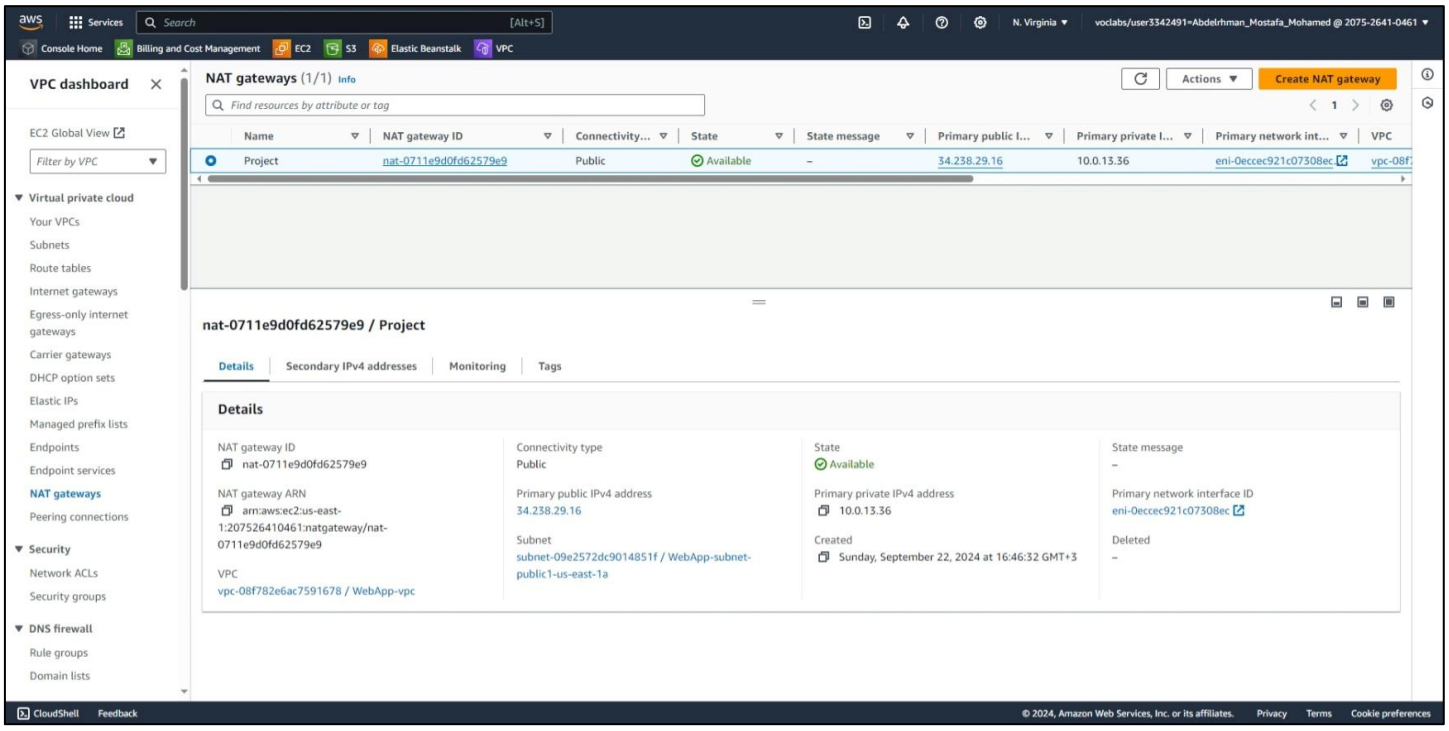
  This setup assumes no upfront payments and is based on on-demand pricing.

Phase 2: Creating a basic functional web application

Task 1: Creating a virtual network







Security Groups File Link

1) **EC2-SG (Security Group for EC2 Instances):**

- **Description**: This security group is used for EC2 instances, allowing inbound SSH (port 22) and HTTP (port 80) access. It is ideal for controlling remote access and web traffic to the EC2 instances.
- **Inbound Rules**: 3 rules, likely including SSH, HTTP, and another service-specific rule.
- **Outbound Rules**: 1 rule, usually allowing all outbound traffic.

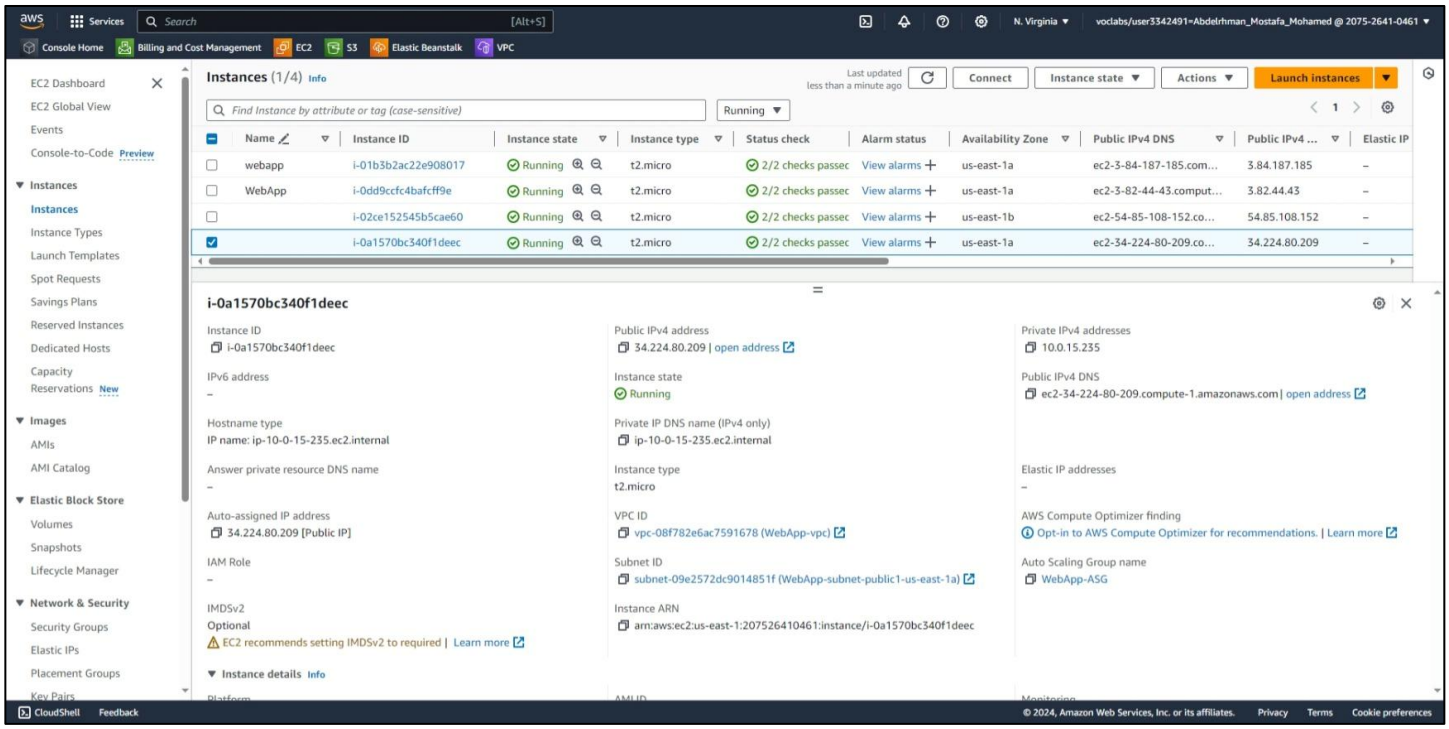
2) **DB-SG (Database Security Group):**

- **Description**: This security group is designed for database instances, allowing inbound MySQL traffic on port 3306, ensuring secure database access.
- **Inbound Rules**: 1 rule for MySQL (port 3306) access.
- **Outbound Rules**: 1 rule, likely allowing all outbound traffic.

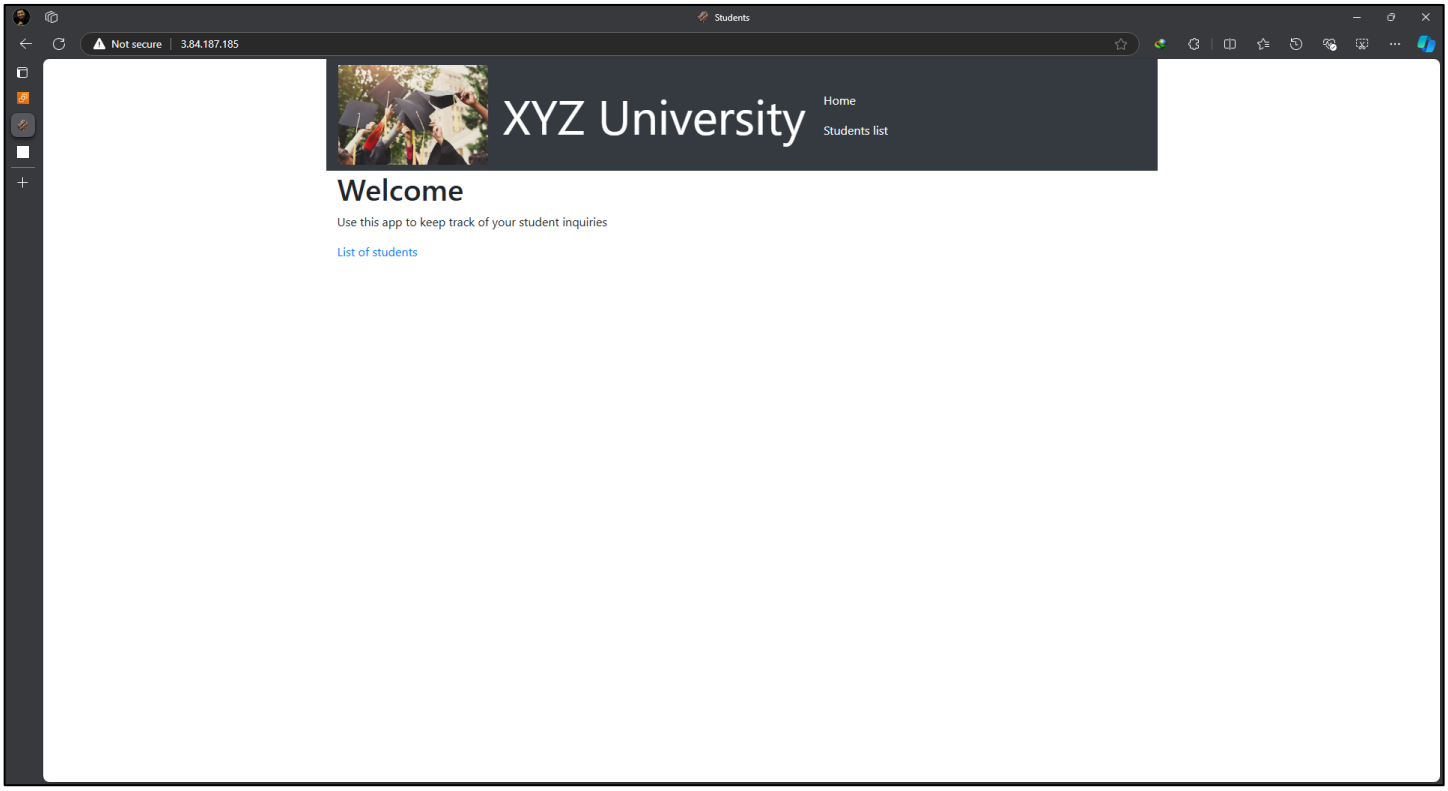
3) **ELB-SG (Security Group for Elastic Load Balancer):**

- **Description**: This security group is for the Elastic Load Balancer (ELB), allowing inbound traffic for load-balanced services such as HTTP and HTTPS.
- **Inbound Rules**: 2 rules, probably for HTTP (port 80) and HTTPS (port 443) traffic.
- **Outbound Rules**: 1 rule, typically allowing all outbound traffic.

Task 2: Creating a virtual machine



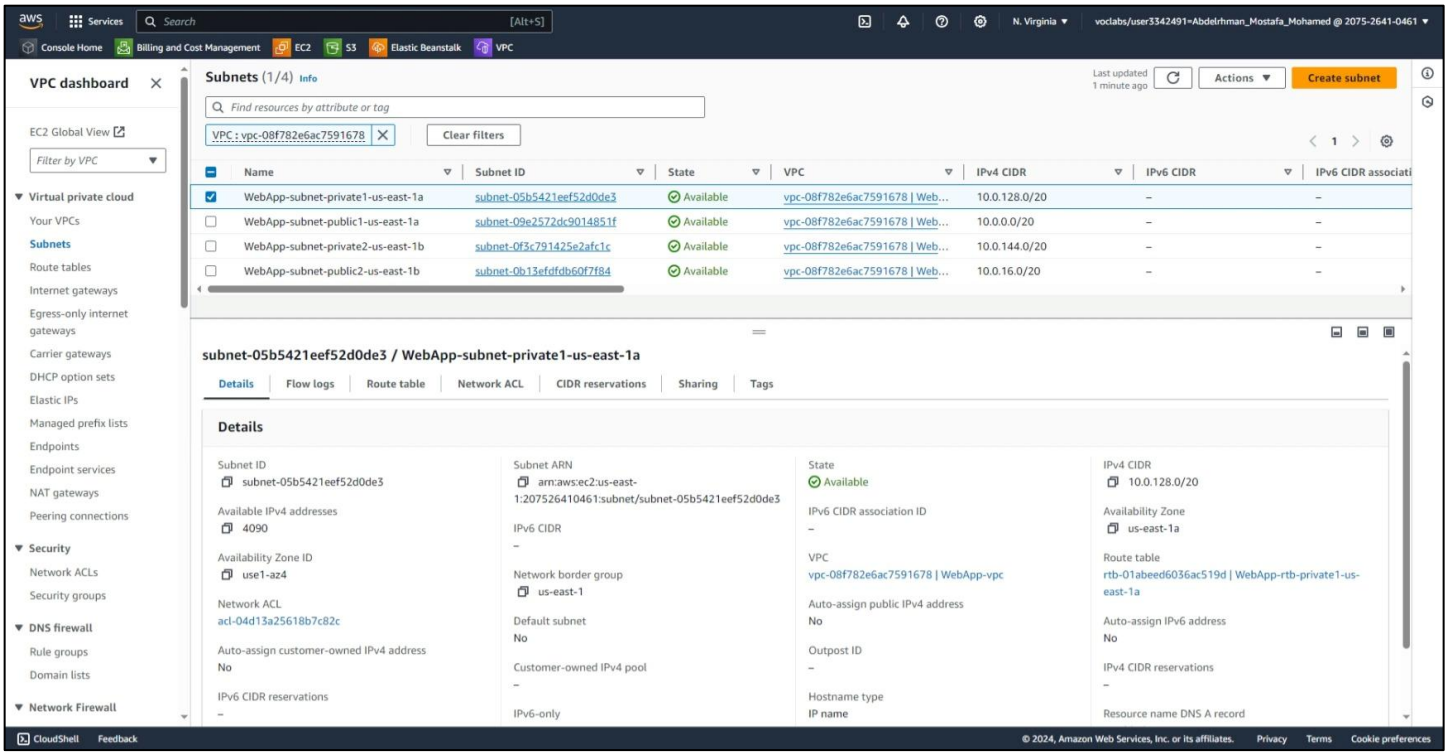
Task 3: Testing the deployment

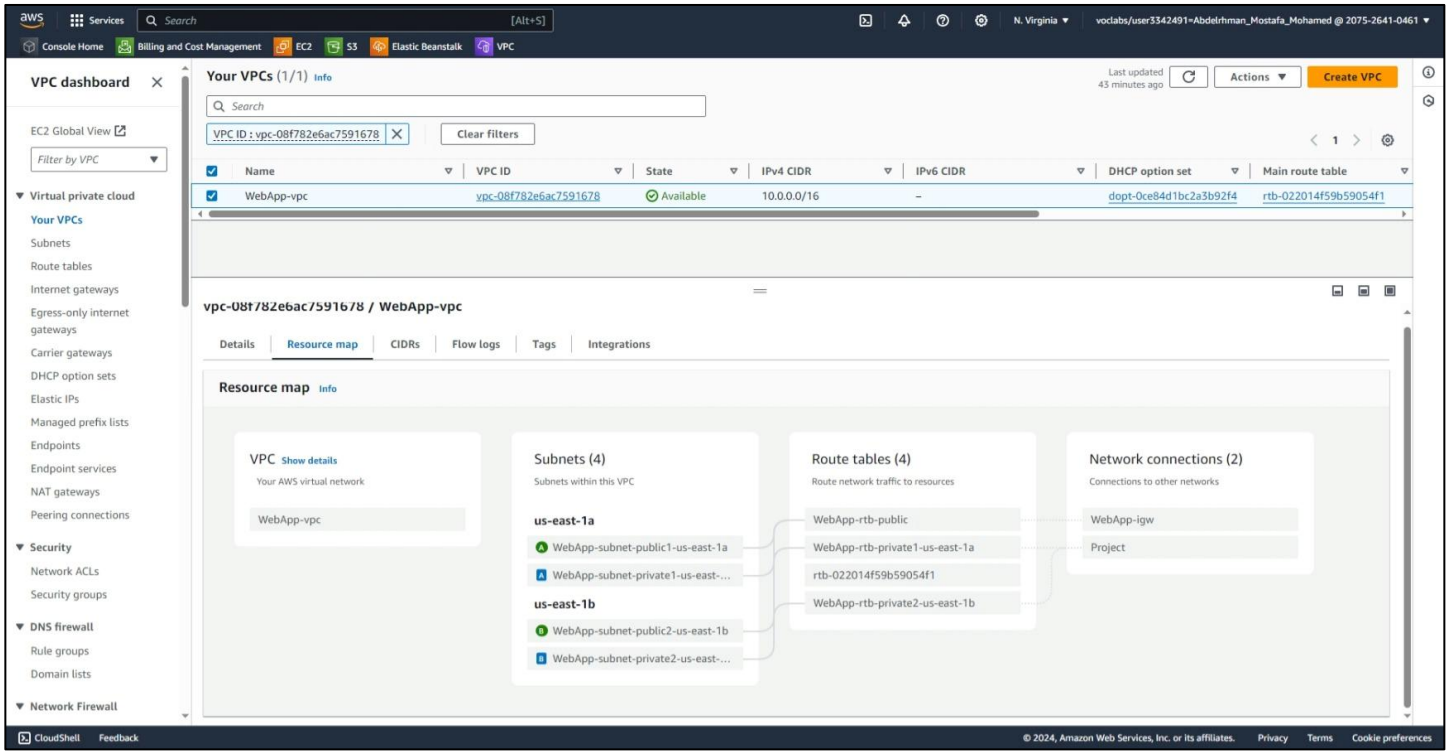


Phase 3: Decoupling the application components

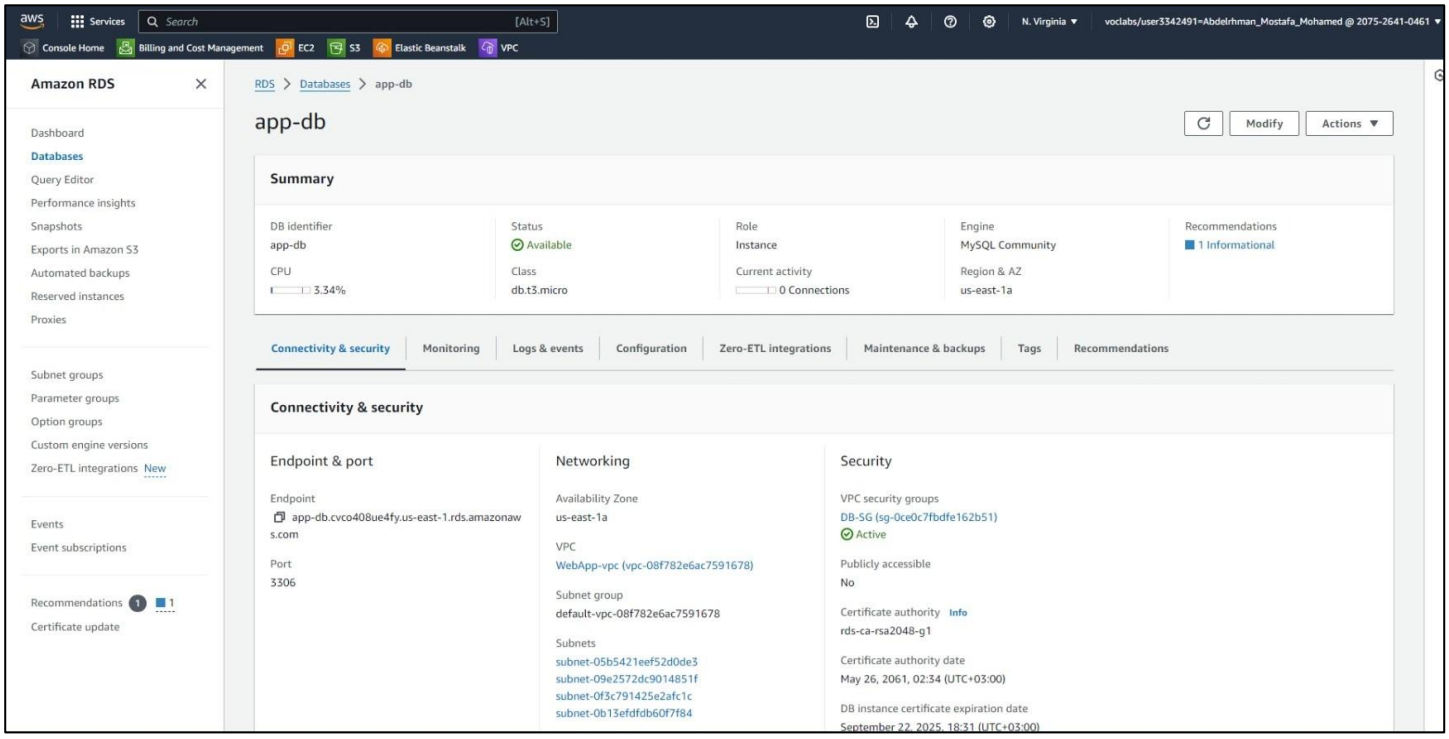
Task 1: Changing the VPC configuration

Note: You need private subnets in a minimum of two Availability Zones.

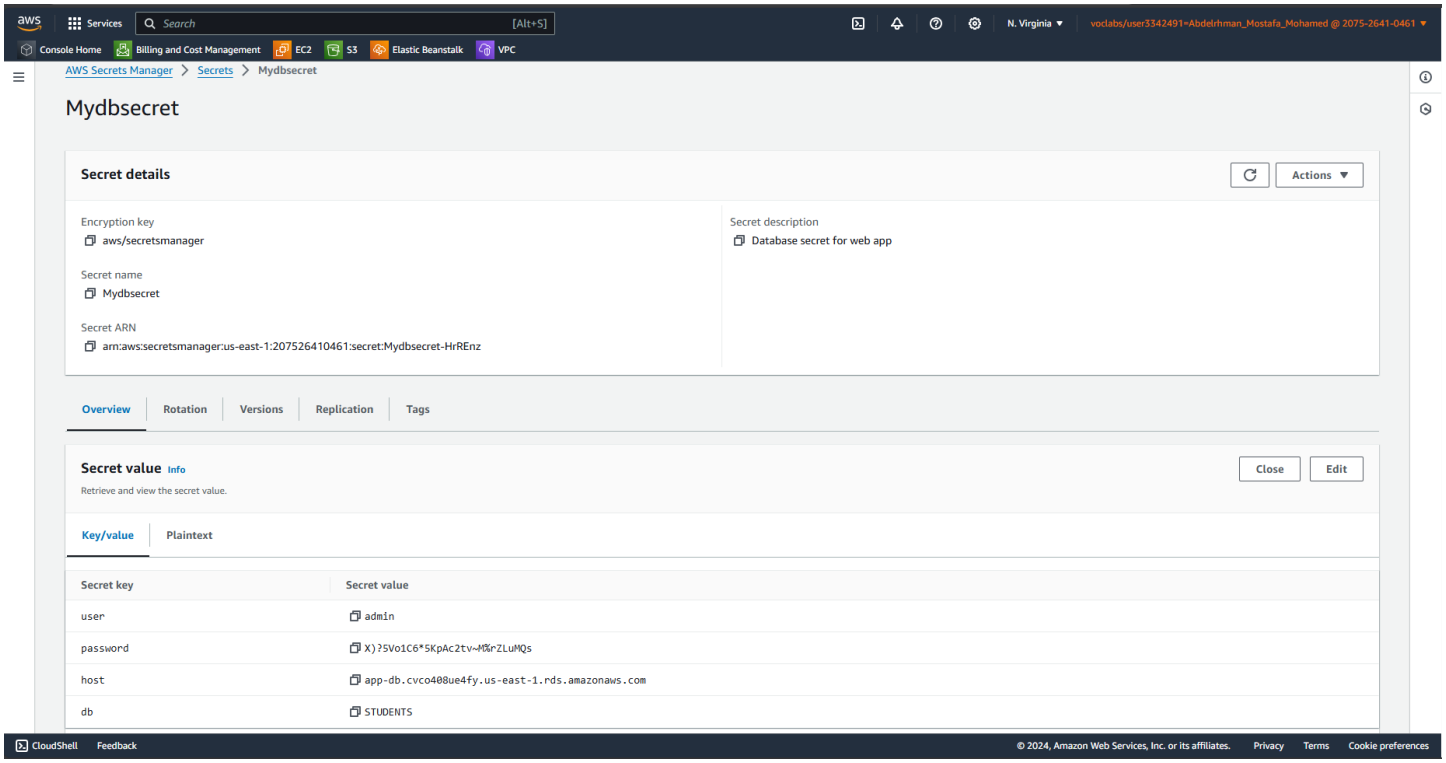




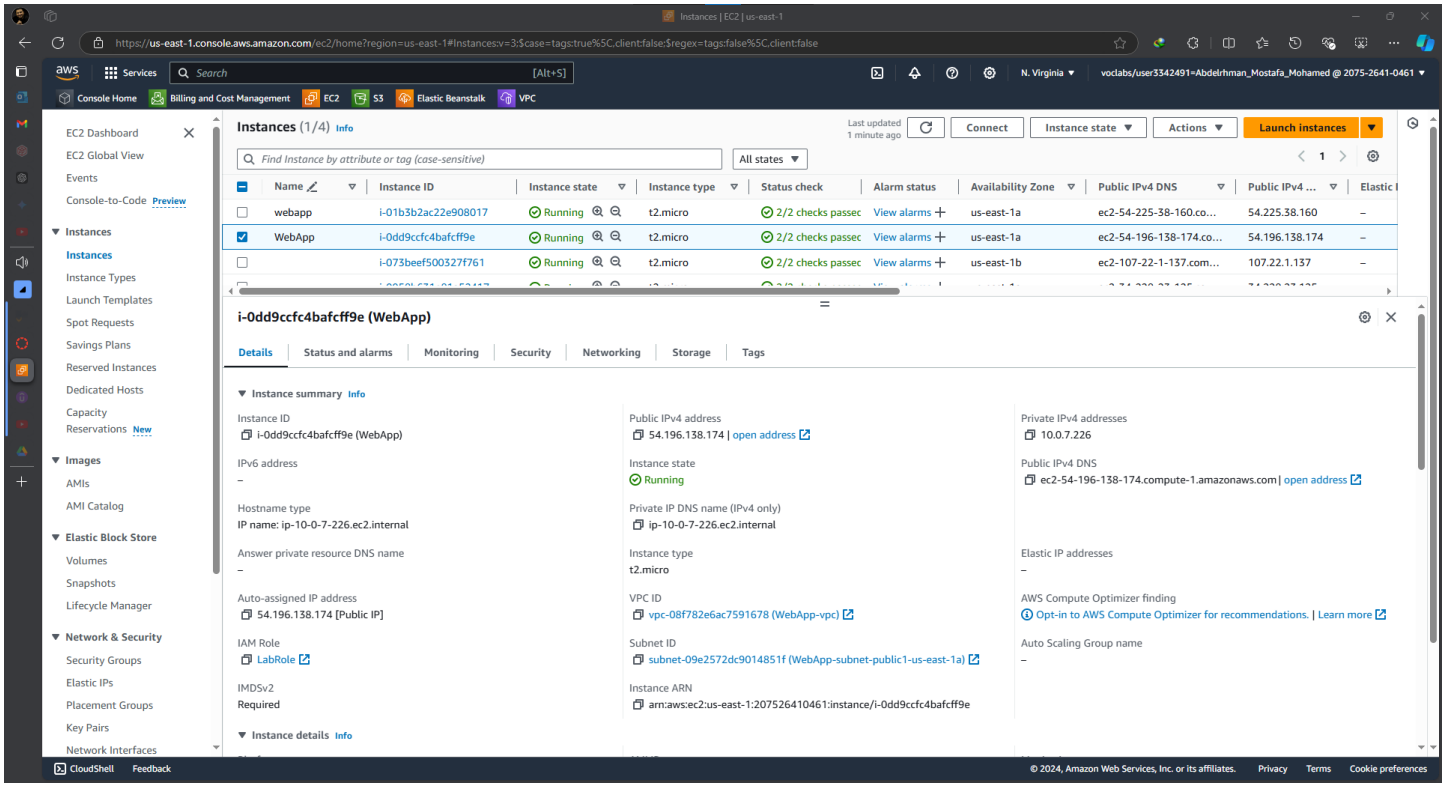
Task 2: Creating and configuring the Amazon RDS database



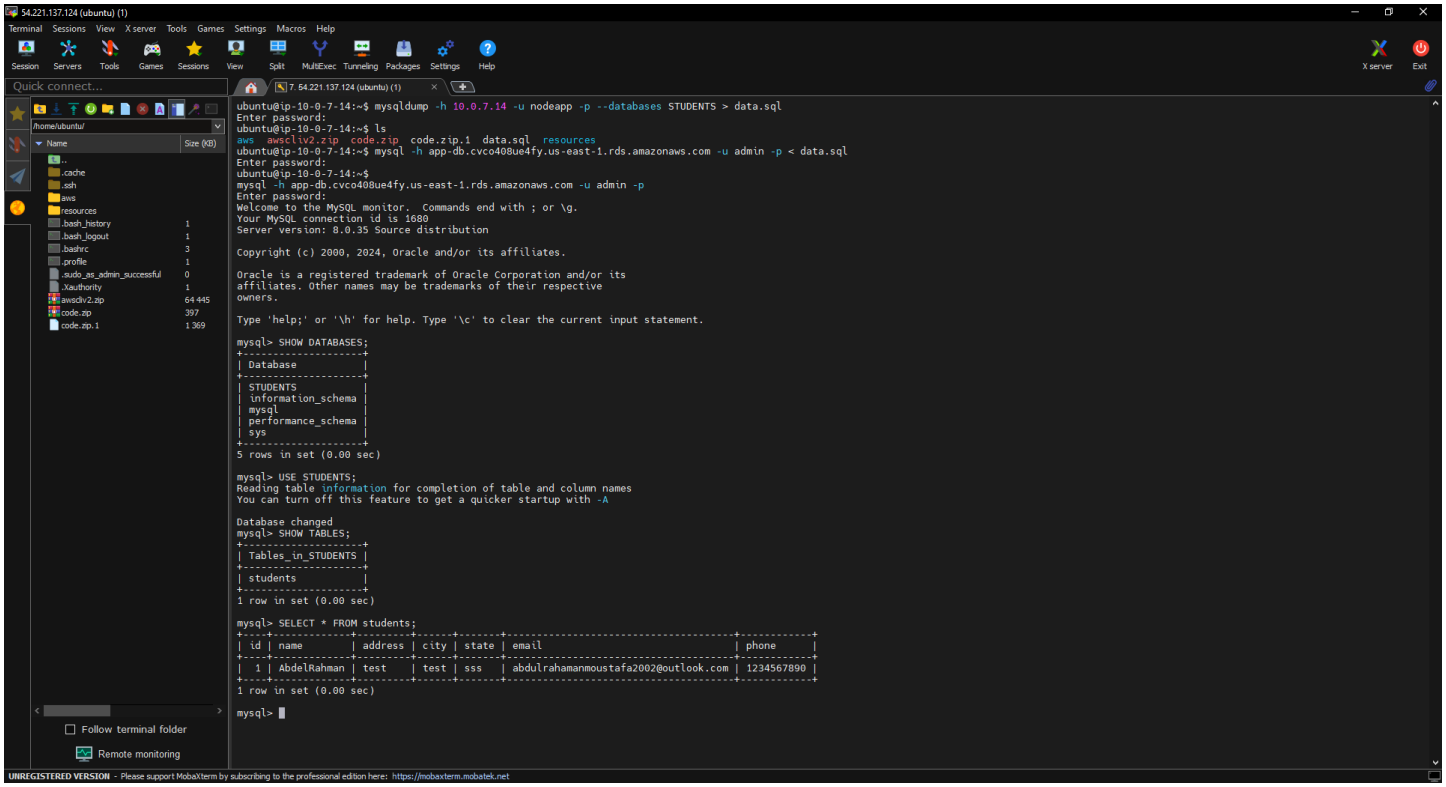
Task 4: Provisioning Secrets Manager

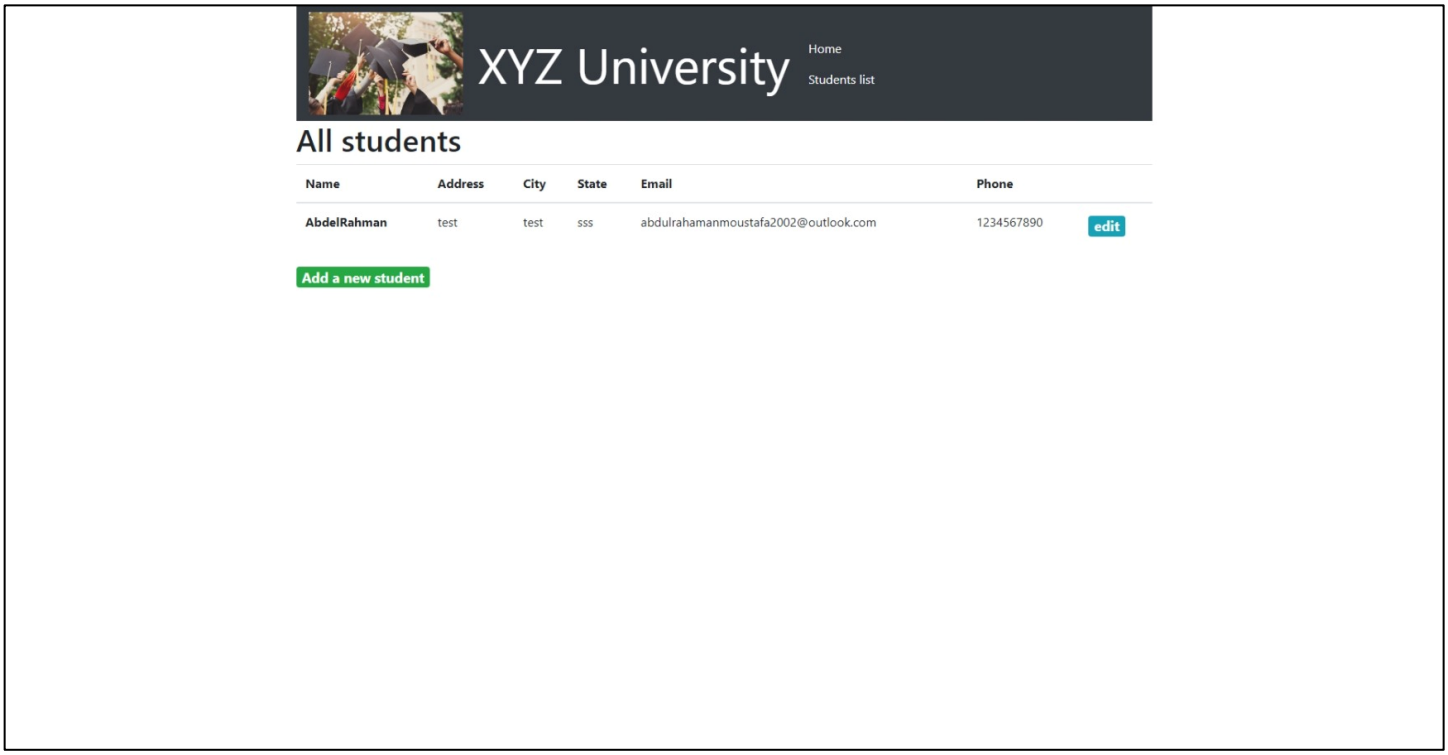


Task 5: Provisioning a new instance for the web server

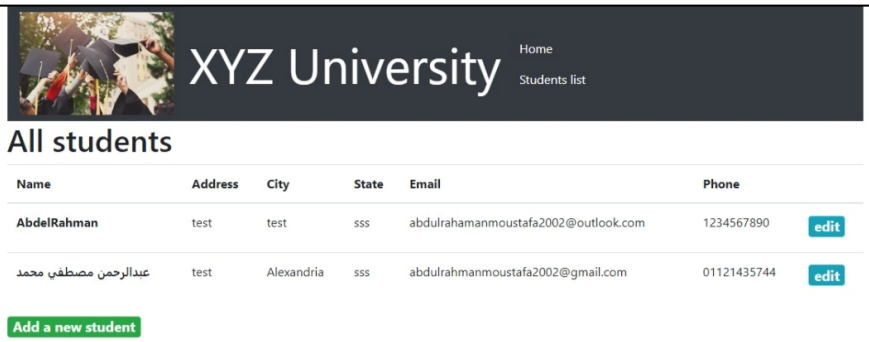


Task 6: Migrating the database



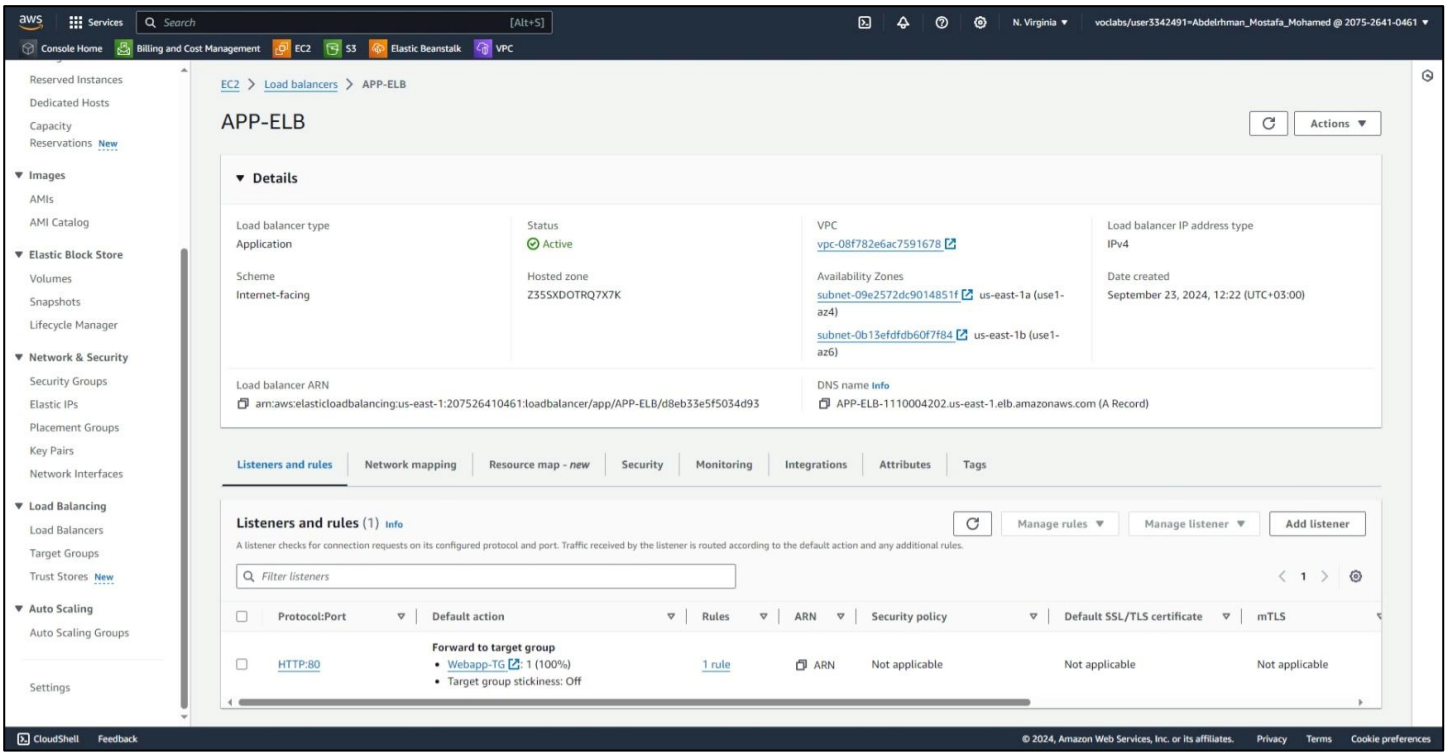


Task 7: Testing the application

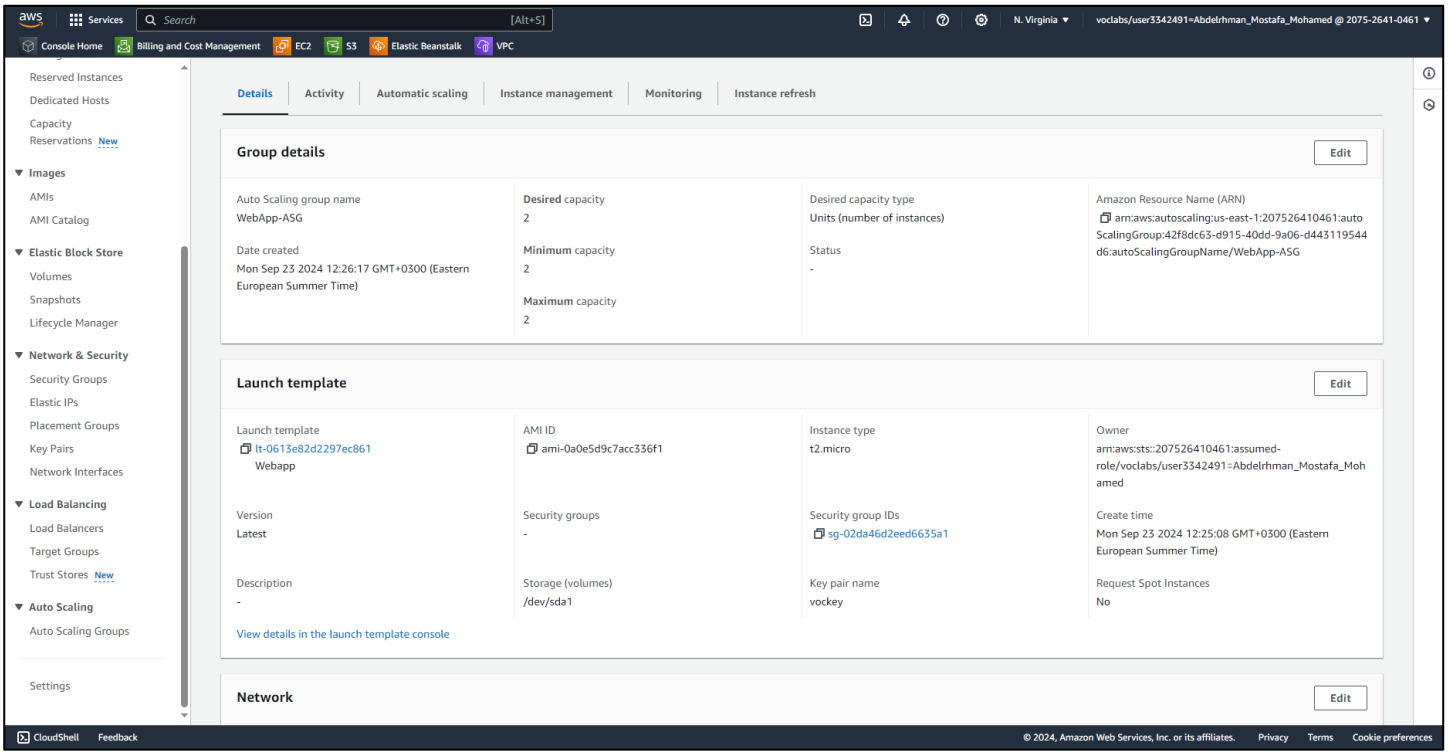


Phase 4: Implementing high availability and scalability

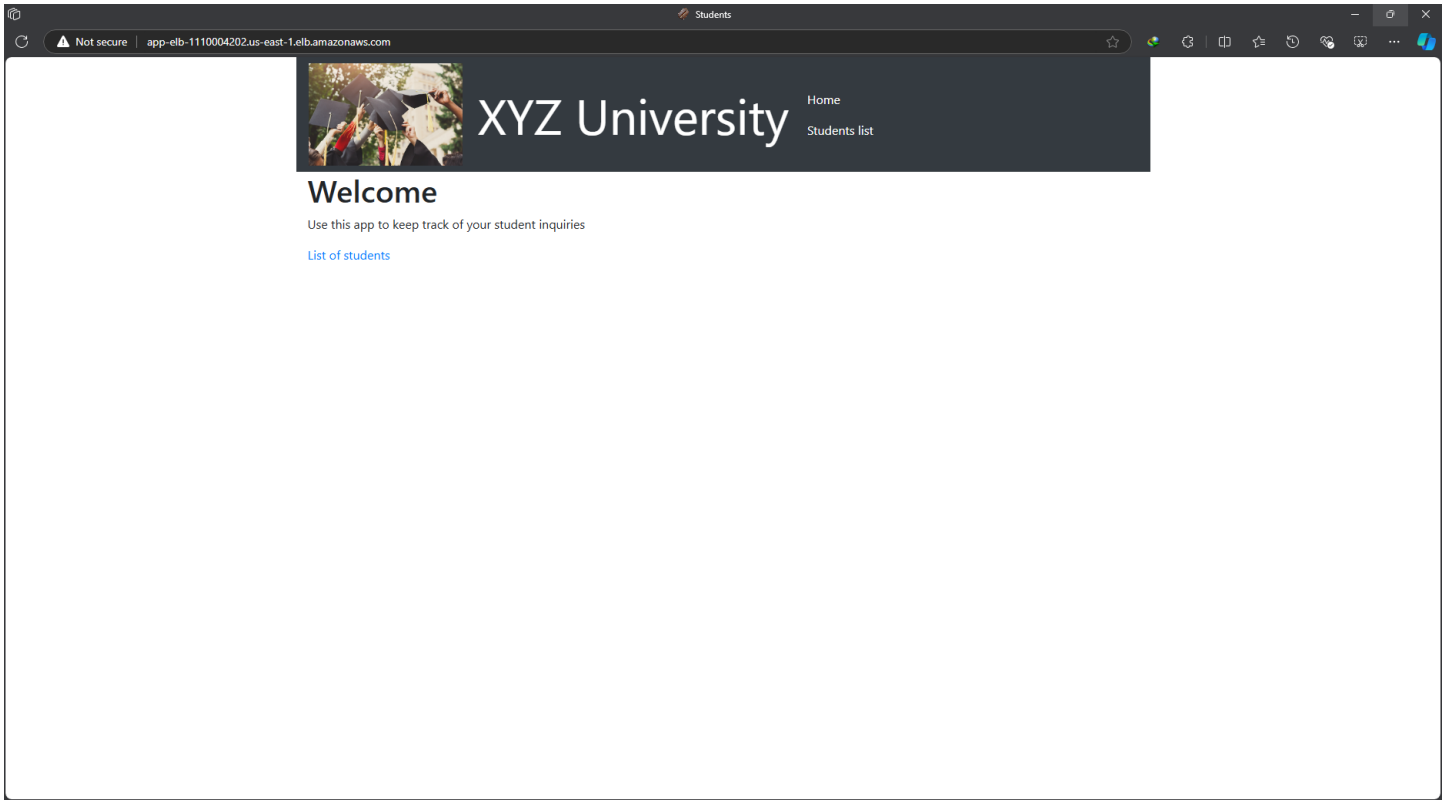
Task 1: Creating an Application Load Balancer



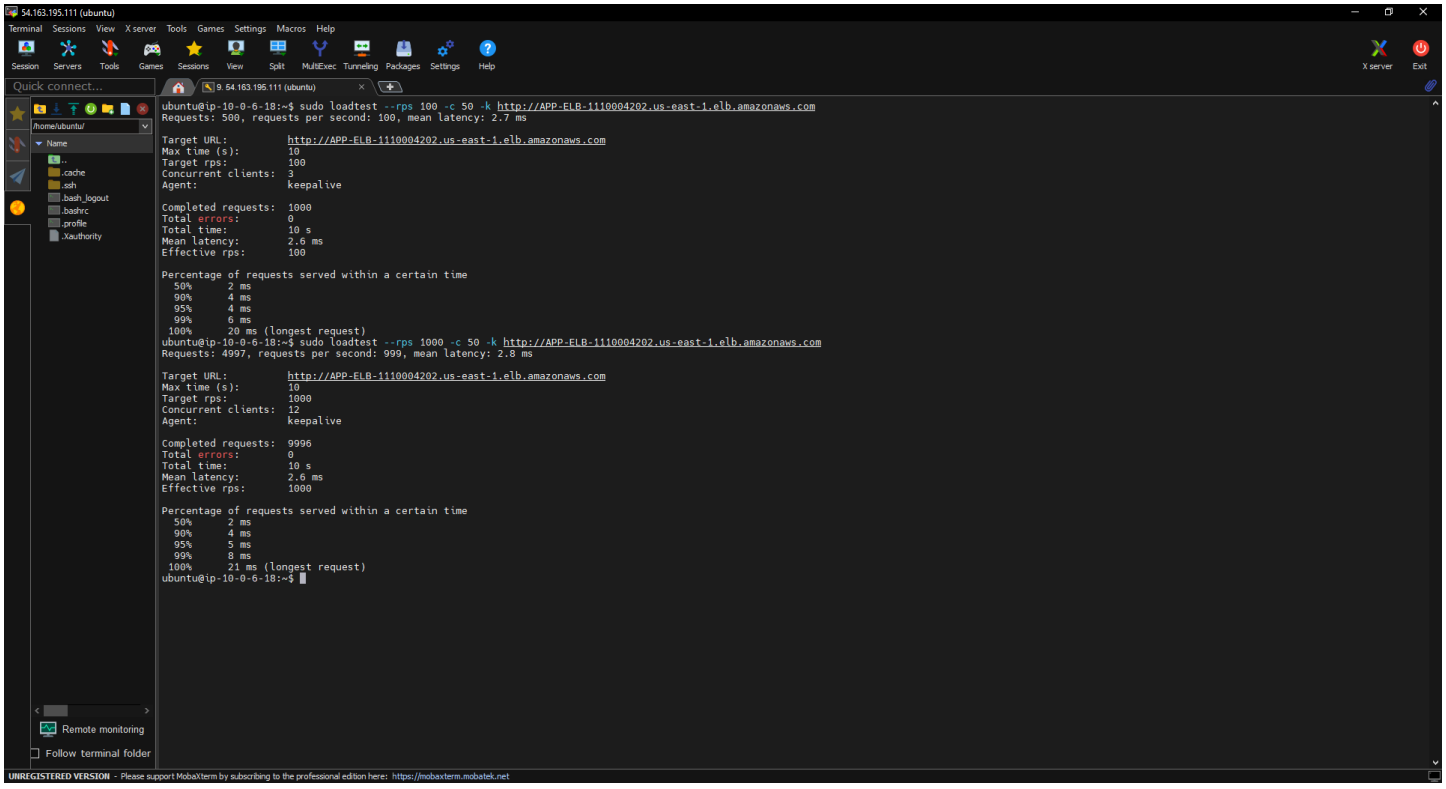
Task 2: Implementing Amazon EC2 Auto Scaling



Task 3: Accessing the application



Task 4: Load testing the application



Badge Link