Design Document

Yiheng Liu (yl5354) *Yiming Zhao (yz4579)*

1 Application Overview

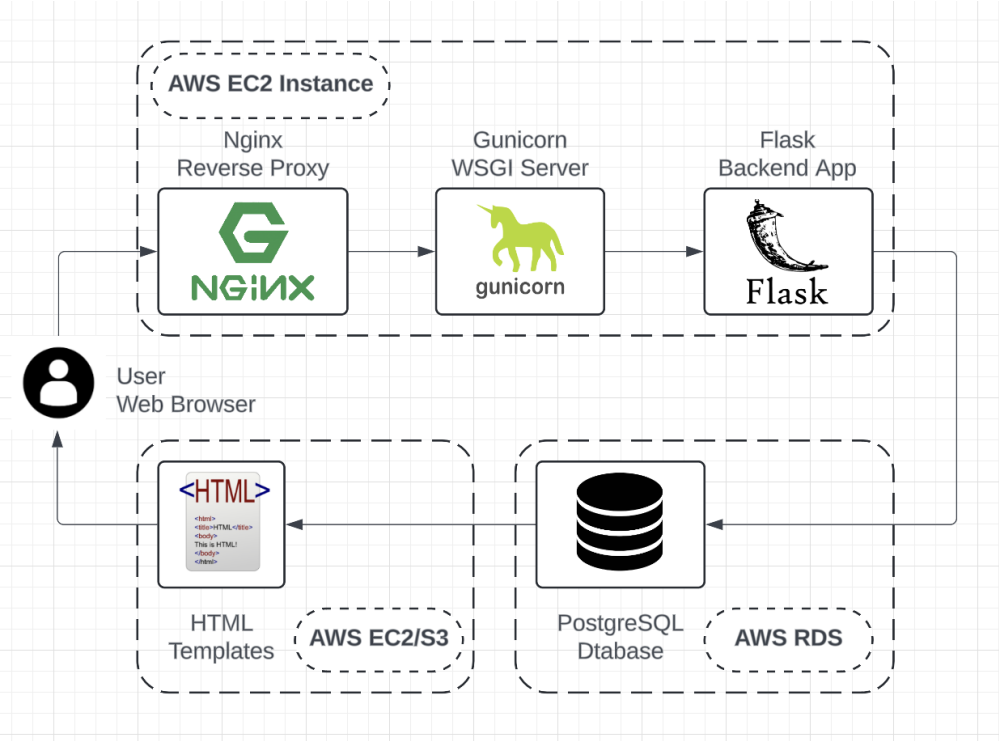

Figure 1: System Overview

This application (http://3.136.117.103:8111/) serves as a comprehensive platform for managing the trading activities and products of the University bookstore. It empowers administrators with tools to monitor, update, and streamline store operations while maintaining a user-friendly interface.

2 Cloud Database

AWS RDS setup

- Access the AWS RDS Console to configure a new PostgreSQL database:

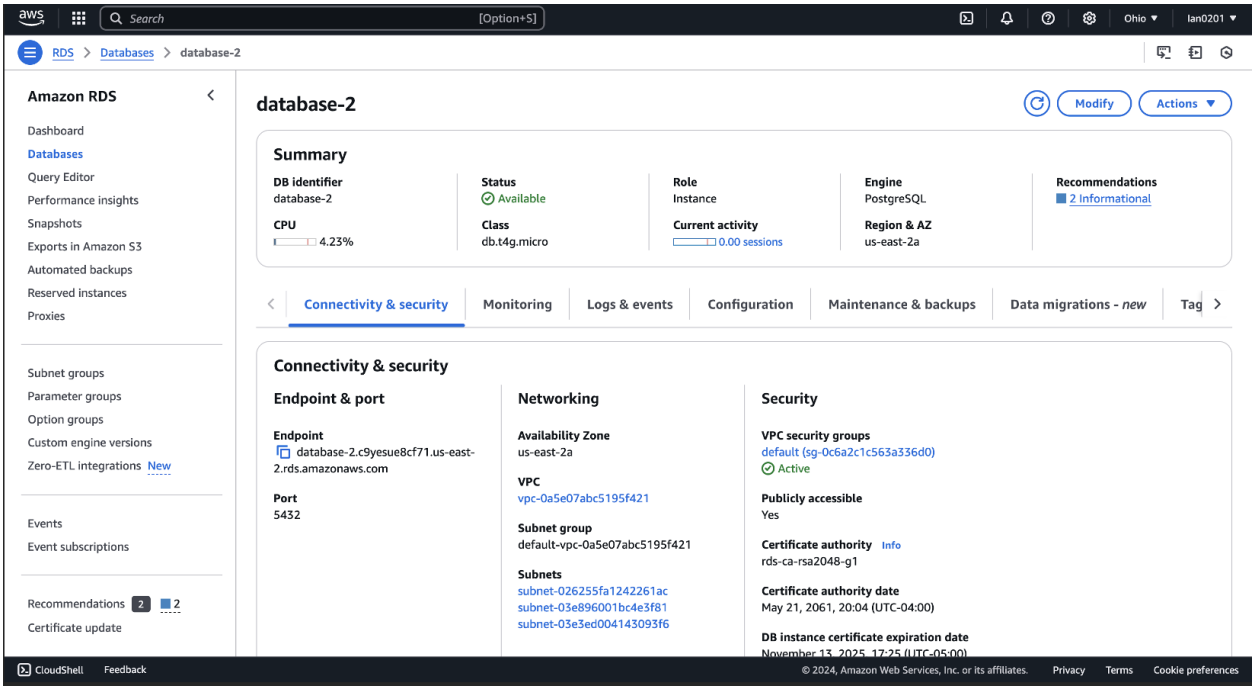

Figure 2: Database Construction

  - **Instance Class:** `db.t2.micro` (Free tier eligible).
  - **Storage:** Minimal allocation to optimize cost.
  - **Credentials:** Specify a robust master username and password.

- Ensure the database is publicly accessible for development, with security groups restricting IP-based access.

Table and Relation Configuration

- Use SQL Pro Studio or equivalent tools to connect to the RDS database with appropriate credentials.
- Define tables and relationships with PostgreSQL commands, incorporating constraints and references to maintain data consistency.

3 Backend Development (Flask)

Project Initialization

- Establish a Python virtual environment and install required dependencies:

```
pip install flask flask_sqlalchemy flask_bcrypt
   flask_jwt_extended
```

- Develop API endpoints for user registration, login, and CRUD operations for data management.

Database Integration

- Set up the database URI using the RDS endpoint:

```
DATABASEURI =
   "postgresql://ELENDB:Ipromise12345@database-2.
c9yesue8cf71.us-east-2.rds.amazonaws.com:5432/ELEN"
```

- Implement SQLAlchemy for seamless database interactions.

API Testing

- Validate API functionality using tools like Postman or curl to simulate client requests.

4 Frontend Development

The front-end interface is crafted using HTML templates and styled with CSS to ensure accessibility and usability for administrative users. Responsive design principles are applied to enhance user experience across devices.

5 Deployment on AWS EC2

EC2 Instance Setup

1. **Instance Initialization:**
   - Launch an Ubuntu-based EC2 instance under the free tier and connect via SSH:

```
ssh -i "<key_pair>.pem" ubuntu@<ec2_public_ip>
```

2. **Environment Preparation:**
   - Update the system and install essential packages:

```
sudo apt update
sudo apt install python3-pip python3-dev libpq-dev
postgresql postgresql-contrib nginx curl
```

3. **Application Deployment:**
   - Clone the GitHub repository containing the project code:

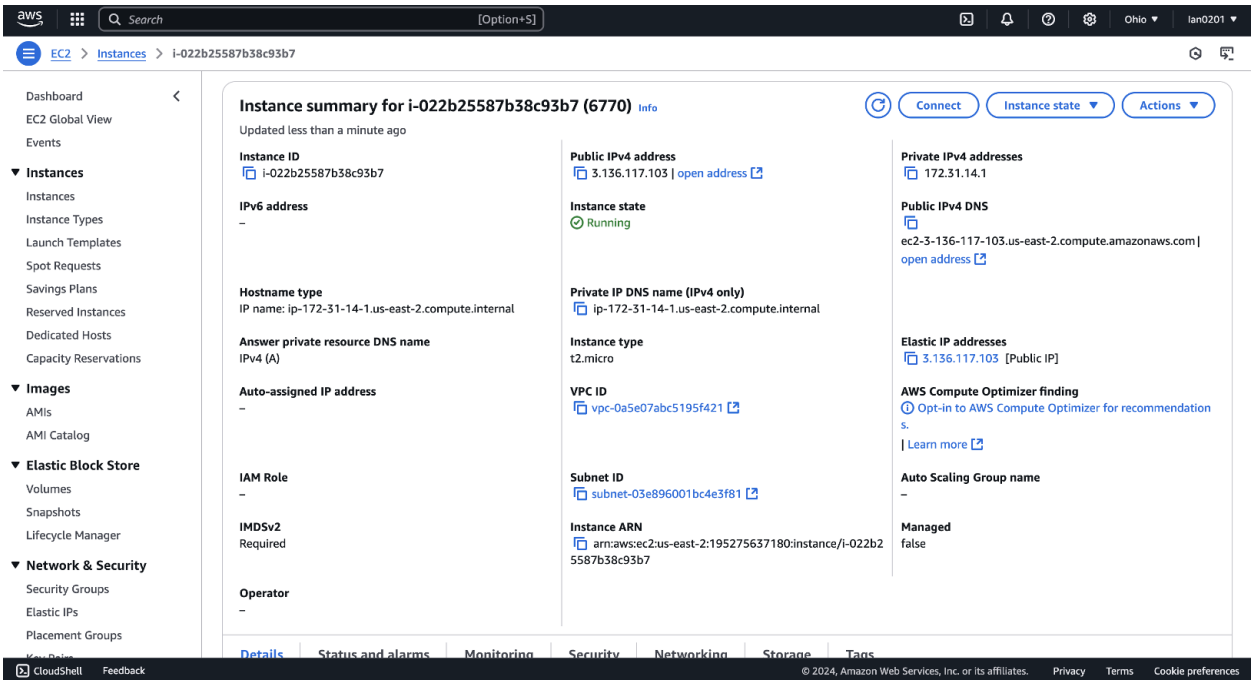

Figure 3: EC2 Instance Setup

```
git clone https://github.com/IAN0201
/6770-Project-Bookstore-Management.git
```

- Install application dependencies:

```
pip install -r requirements.txt
```

- Run the application locally to confirm functionality before deployment.

Production Configuration

1. **Gunicorn Setup:**

   - Define a systemd service file for Gunicorn:

```
sudo nano /etc/systemd/system/gunicorn.service
```

   Add the following configuration:

```
[Unit]
Description=Gunicorn instance to serve Flask app
After=network.target

[Service]
User=ubuntu
Group=www-data
```

```
WorkingDirectory=/home/ubuntu/apiv1
Environment="PATH=/home/ubuntu/apiv1/env/bin"
ExecStart=/home/ubuntu/apiv1/env/bin/gunicorn -w 4 -k
uvicorn.workers.UvicornWorker -b 0.0.0.0:8111
   server:asgi_app

[Install]
WantedBy=multi-user.target
```

- Enable and start the service to ensure persistent availability:

```
sudo systemctl start gunicorn
sudo systemctl enable gunicorn
```

2. **Nginx Configuration:**

- Configure Nginx to act as a reverse proxy for Gunicorn:

```
sudo nano /etc/nginx/sites-available/api
```

Insert the following block:

```
server {
    listen 80;
    server_name <server_ip>;

    location / {
        proxy_pass http://127.0.0.1:8111;
    }
}
```

- Test the configuration and restart the Nginx service:

```
sudo nginx -t
sudo systemctl restart nginx
```

3. **Elastic IP Allocation:**

- Allocate an Elastic IP to ensure a consistent public-facing address for the application.

Accessing the Application

Access the fully deployed application using the public IP: http://3.136.117.103:8111/