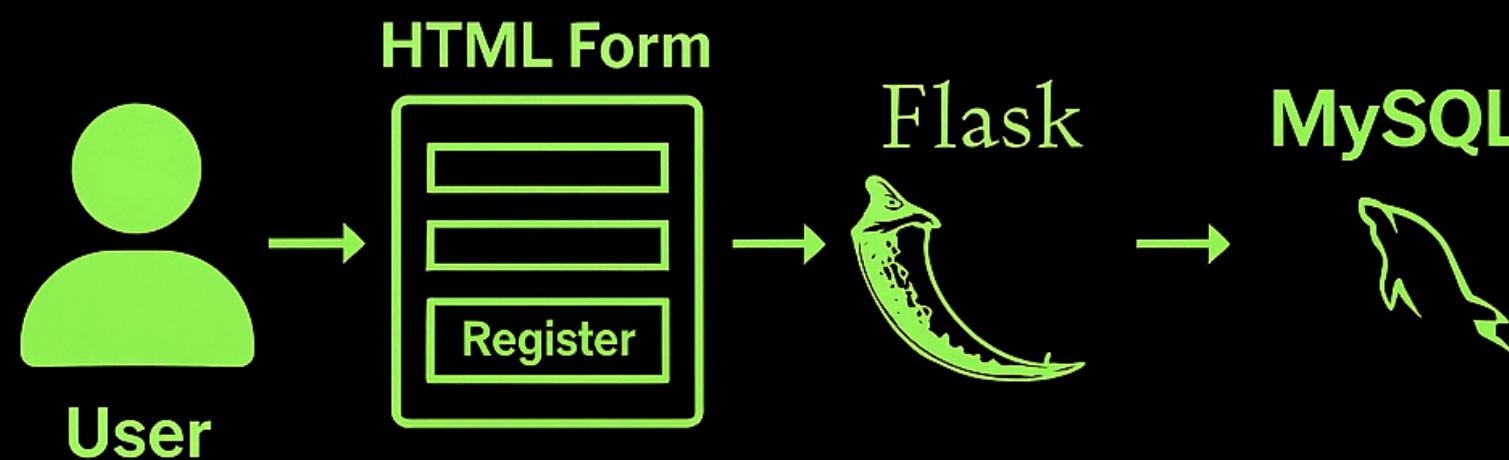


CLOUD-BASED USER MANAGEMENT SYSTEM

USING AWS
EC2 + RDS + PYTHON
FLASK

PROJECT GOAL



OBJECTIVE :

- Develop a user registration and management system
- Host the application on AWS EC2
- Store user data in AWS RDS utilizing MySQL
- Facilitate user registration through an HTML form
- Enable administrators to search for users via a web interface

ABOUT TECHNOLOGY



Flask



aws

Python

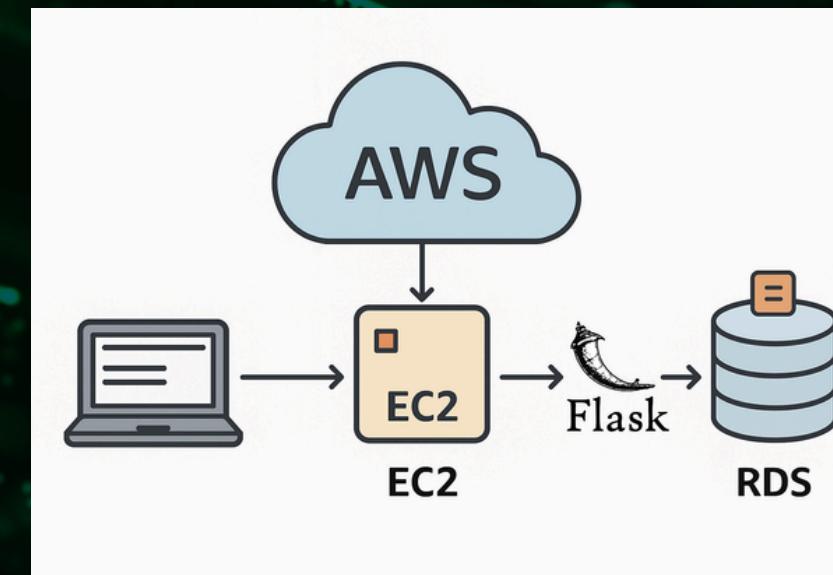


POSTMAN

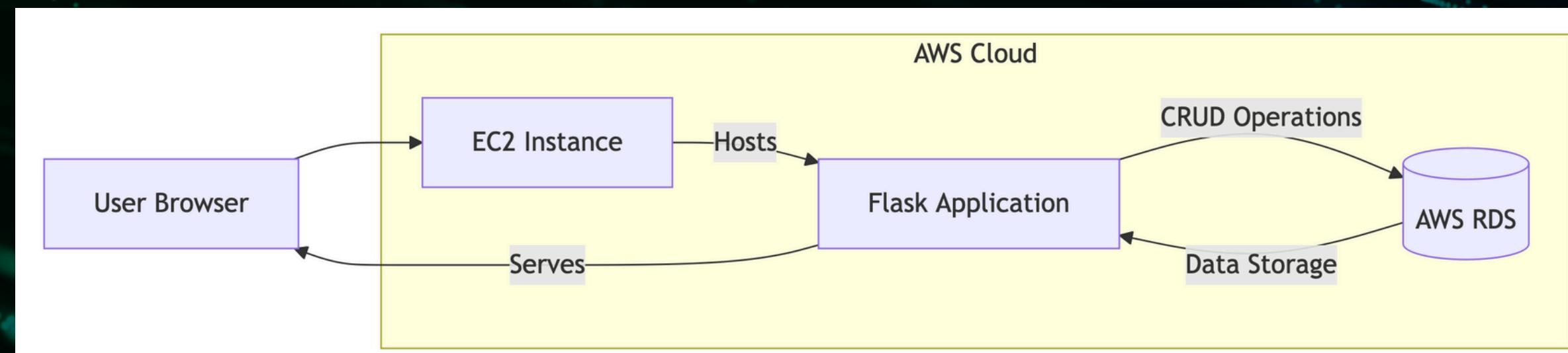
Tech Stack

- **Frontend:** HTML, CSS
- **Backend:** Python + Flask
- **Database:** AWS RDS (MySQL)
- **Cloud Services:** AWS EC2, RDS
- **Tools:** Postman

ARCHITECTURE OVERVIEW

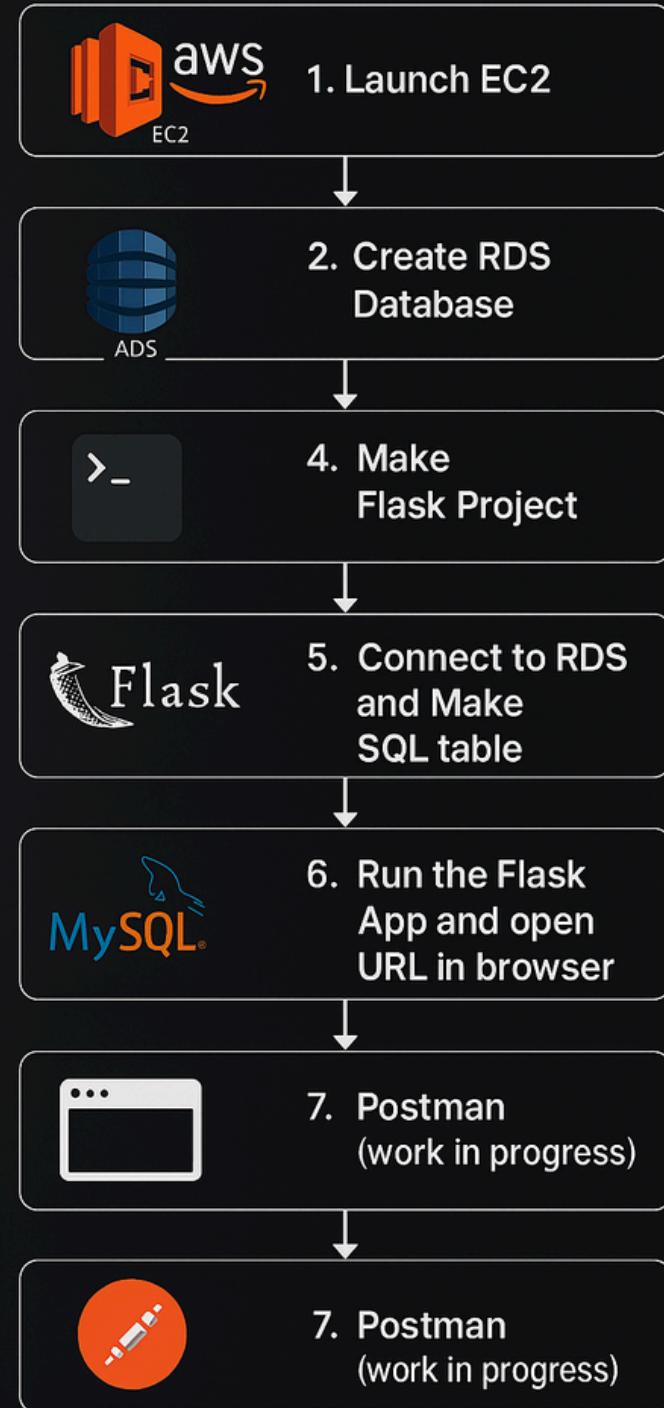


- User submits form via browser (HTTPS)
- Request routed to Flask app on EC2
- Flask processes logic/validations
- CRUD operations executed on RDS
- RDS returns data to Flask
- Flask renders response to user
- HTML content displayed in browser



PROJECT SETUP STEPS

Project Setup Steps



```
Last login: Fri Jul 18 10:11:35 on ttys000
jeetpatel@Jeets-MacBook-Air ~ % chmod 400 ~/Desktop/flask_key.pem
ssh -i ~/Desktop/flask_key.pem ubuntu@3.18.225.240

The authenticity of host '3.18.225.240 (3.18.225.240)' can't be established.
ED25519 key fingerprint is SHA256:Qd20YBdy5KdXX2cs00FOa7Riv5E+wGQdoufi7ZGBUUI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.18.225.240' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Jul 18 17:06:04 UTC 2025

System load:  0.0          Temperature:      -273.1 °C
Usage of /:   25.3% of 6.71GB  Processes:        106
Memory usage: 24%           Users logged in:   0
Swap usage:   0%            IPv4 address for ens5: 172.31.46.231

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-231:~$
```

```
from flask import Flask, render_template, request
import mysql.connector

app = Flask(__name__)

# MySQL RDS connection configuration
db = mysql.connector.connect(
    host="userdb.ch62kqu8cwq7.us-east-2.rds.amazonaws.com",
    user="admin",
    password="Patel2025",
    database="userdb"
)

cursor = db.cursor()

@app.route('/')
def home():
    return "Welcome to the User Management System"

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        # Get form data
        username = request.form['username']
        password = request.form['password']
        first_name = request.form['first_name']
        last_name = request.form['last_name']
        email = request.form['email']

        # Insert into database
        query = "INSERT INTO users (username, password, first_name, last_name, email) VALUES (%s, %s, %s, %s, %s)"
        values = (username, password, first_name, last_name, email)

        cursor.execute(query, values)
        db.commit()

        return 'User registered successfully!'

    return render_template('register.html')

if __name__ == '__main__':
    app.run(debug=True)
```

User_Database_Schema

```
CREATE DATABASE user_db;  
USE user_db;  
  
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(100),  
    username VARCHAR(50),  
    password VARCHAR(100)  
);
```

Admin Panel - Search Users

```
<h2>Admin Panel – Search Users</h2>  
<form method="GET">  
    <input name="keyword" placeholder="Search by name, email, or username">  
    <input type="submit" value="Search">  
</form>  
  
<ul>  
    {% for user in users %}  
        <li>{{ user[1] }} | {{ user[2] }} | {{ user[3] }}</li>  
    {% endfor %}  
</ul>
```

EC2 Instances (running)

Instances (1)				
	Name	Instance ID	Instance type	Instance state
<input checked="" type="checkbox"/>	ip-172-31-90118	i-08fa46488df88a7	t2.micro	running

RDS Databases

Databases				
	DB Identifier	Engine	Region and AZ	DB instance class
<input checked="" type="radio"/>	userdb MySQL-9.0.36	MySQL	us-east-1a	db.t3.micro
<input type="radio"/>	userdb	MySQL	0	db.t3.micro

User Registration Form

```
<h2>User Registration</h2>  
<form method="POST">  
    Name: <input name="name"><br>  
    Email: <input name="email"><br>  
    Username: <input name="username"><br>  
    Password: <input name="password" type="password"><br>  
    <input type="submit" value="Register">  
</form>
```

USER FUNCTIONALITY

The screenshot shows a user login form with the following elements:

- Login**: The title of the form.
- You have been logged out.**: A green success message.
- Username or Email**: An input field containing "Tushy101".
- Password**: An input field showing masked text "*****".
- Login**: A blue button.
- Don't have an account? Register**: A link at the bottom.

Route: /register

Fields:

Username, Password,
First Name, Last Name,
Email

Features:

Validate input

Insert into users table

Retrieve/Search/Delete users

BACKEND CODE (FLASK APP.PY)

```
from flask import Flask, render_template, request, jsonify, redirect, url_for
import mysql.connector

app = Flask(__name__)

# MySQL RDS connection config
db_config = {
    'host': 'userdb.ch62kqu8cwq7.us-east-2.rds.amazonaws.com',
    'user': 'admin',
    'password': 'Patel2025',
    'database': 'userdb'
}

def get_db_connection():
    return mysql.connector.connect(**db_config)

@app.route('/')
def index():
    return render_template('simple_index.html')

@app.route('/login')
def login():
    return render_template('simple_login.html')

@app.route('/register', methods=['GET'])
def register_page():
    return render_template('simple_register.html')

@app.route('/register', methods=['POST'])
def register_user():
    data = request.form
    conn = get_db_connection()
    cursor = conn.cursor()
    try:
        query = """
            INSERT INTO users (username, password, first_name, last_name, email)
            VALUES (%s, %s, %s, %s, %s)
        """
        values = (
            data['username'],
            data['password'], # Note: hash passwords in production
            data['first_name'],
            data['last_name'],
            data['email']
        )
        cursor.execute(query, values)
        conn.commit()
        return "✅ User registered successfully!"
    except mysql.connector.Error as err:
        return f"❌ Error: {err}"
    finally:
        cursor.close()
        conn.close()
```

API TESTING WITH POSTMAN

The screenshot shows the Postman application interface. At the top, there is a header bar with a 'POST' button, the URL 'http://3.18.225.240:5000/register', and a 'Send' button. Below the header, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is selected and contains an 'x-www-form-urlencoded' section with four fields: 'username' (value: 'johndeer'), 'email' (value: 'johnDeer@example.com'), 'password' (value: 'jd'), and 'first name' (value: 'John'). Below the body section, there are buttons for 'Body', 'Cookies', 'Headers (7)', and 'Test Results'. On the right side of the interface, there is a preview window showing a 'Login' form with a green success message: 'Registration successful! Please login.' Below the message are input fields for 'Username or Email' and 'Password', and a 'Login' button.

Test POST /register with form data

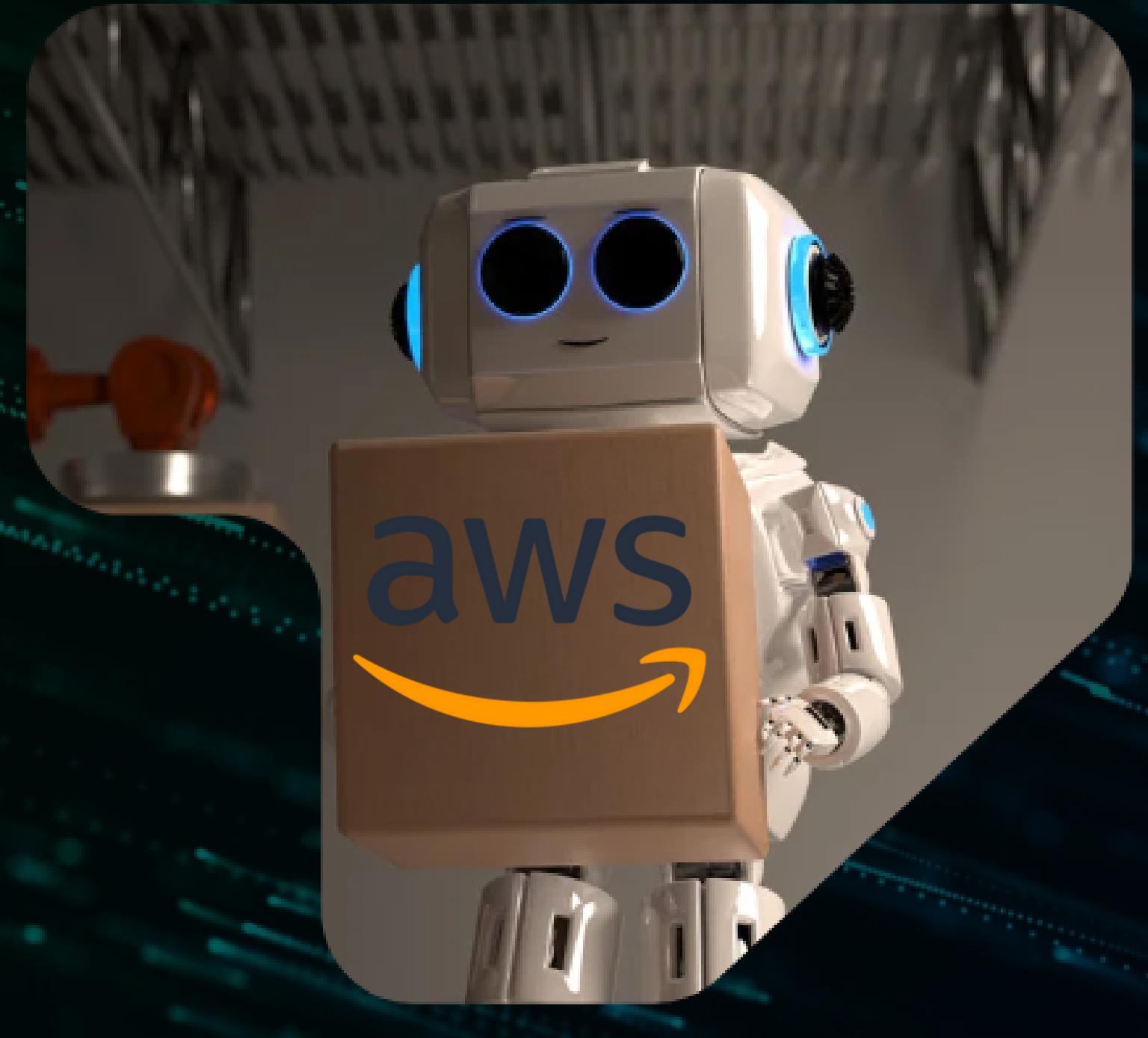
Test GET /search?q=keyword

Test DELETE /delete/<user_id>

Correct response codes

DELIVERABLES

- EC2-hosted Flask Web App
- Registration + Admin HTML Pages
 - MySQL RDS Database



THANK YOU

...

...