🚀 Full Stack Deployment: Flask App on EC2 with AWS RDS (MySQL) Backend

🎯 Objective

Deploy a Python Flask-based student registration web app on an Amazon EC2 instance, connect it to a MySQL database hosted on Amazon RDS, and make it accessible over the internet.

🎁 Project Stack

Layer	Technology
Frontend	HTML (in Flask templates)
Backend	Python (Flask)
Database	Amazon RDS (MySQL)
Hosting	Amazon EC2
Git Hosting	GitHub

✅ Step-by-Step Deployment Guide

✅ STEP 1: Login into EC2

```
bash
CopyEdit
ssh -i my-key.pem ec2-user@3.108.222.101
```

🔍 Explanation:

- `ssh`: Secure shell for connecting to EC2
- `-i`: Specifies your private key file
- `ec2-user`: Default user for Amazon Linux (use `ubuntu` for Ubuntu)
- `3.108.222.101`: Replace with your EC2 Public IPv4

✅ STEP 2: Update & Install Required Packages

Amazon Linux:

```
bash
CopyEdit
sudo yum update -y
sudo yum install python3 git -y
```

Ubuntu:

```
bash
CopyEdit
sudo apt update && sudo apt upgrade -y
sudo apt install python3 python3-pip git -y
```

✅ STEP 3: Clone Flask Project from GitHub

```
bash
CopyEdit
git clone https://github.com/swati-zampal/stud-reg-flask-app.git
cd stud-reg-flask-app
```

✅ STEP 4: Set Up Virtual Environment

```
bash
CopyEdit
python3 -m venv venv
source venv/bin/activate
```

✅ STEP 5: Install Flask and PyMySQL

```
bash
CopyEdit
pip install flask pymysql
```

☁️ Amazon RDS (MySQL) Setup – Step-by-Step

🔷 STEP 6.1: Create RDS Instance

- Go to AWS RDS Console → **Create Database**
- Choose **Standard Create**
- Engine: **MySQL**
- Version: Latest
- Template: **Free Tier**

- DB Identifier: `studentdb`
- Username: `admin`
- Password: `yoursecurepass123`
- Instance Class: `db.t3.micro`
- Public Access: **Yes**
- VPC Security Group: **Create new** (Name: `rds-access-group`)
- Initial DB Name: `studentdb`
- Port: `3306`

◆ STEP 6.2: Configure Inbound Access

Go to:

- **EC2 → Security Groups → rds-access-group → Inbound Rules**
- Add:
  - Type: **MySQL/Aurora**
  - Port: `3306`
  - Source: `0.0.0.0/0` (For production: use EC2's public IP/CIDR)

◆ STEP 6.3: Get RDS Endpoint

Example:

```
CopyEdit
studentdb.abc123xyz.rds.amazonaws.com
```

✅ STEP 6.4: Modify `app.py` for RDS

```
python
CopyEdit
import pymysql

conn = pymysql.connect(
    host="your-rds-endpoint.rds.amazonaws.com",
    user="admin",
    password="yoursecurepass123",
    database="studentdb"
)
```

📌 Save using `Ctrl + O` and exit with `Ctrl + X`.

✅ STEP 7: Create MySQL Table in RDS

```
bash
CopyEdit
nano create_table.py
```

Paste this code:

```
python
CopyEdit
import pymysql

conn = pymysql.connect(
    host="your-rds-endpoint.rds.amazonaws.com",
    user="admin",
    password="yoursecurepass123",
    database="studentdb"
)

cursor = conn.cursor()

cursor.execute("""
CREATE TABLE IF NOT EXISTS students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(20),
    course VARCHAR(50),
    address TEXT
)
""")

conn.commit()
cursor.close()
conn.close()
```

Run it:

```
bash
CopyEdit
python3 create_table.py
```

✅ STEP 8: Allow Flask Access from All IPs

Edit bottom of `app.py`:

```
python
CopyEdit
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

✅ STEP 9: Open Port 5000 in Security Group

- Go to EC2 Dashboard → Your instance → **Security Group → Edit Inbound Rules**
- Add:
    - Type: **Custom TCP**

- Port: `5000`
- Source: `0.0.0.0/0`

✅ STEP 10: Run the Flask Application

```
bash
CopyEdit
python3 app.py
```

✅ Output:

```
csharp
CopyEdit
 * Running on http://0.0.0.0:5000
```

🌐 Visit from browser:

```
cpp
CopyEdit
http://<your-ec2-public-ip>:5000
```

✅ STEP 11: Test the Full Flow

1. Submit the student registration form.
2. See confirmation message.
3. Check RDS via SQL client or write `SELECT * FROM students` in a Python script.

📦 Project Summary

Component	Task
EC2	Hosted the Flask App
GitHub	Cloned the application source code
Python venv	Created isolated development environment
pip	Installed required libraries
RDS	Hosted MySQL Database
Flask App	Connected to RDS and made browser-accessible

📂 Project Links

- 🔗 **GitHub**: https://github.com/shinderushi2363/aws-vpc-flowlogs-to-s3
- 💼 **LinkedIn**: Rushikesh Shinde