Deployment Instructions for Student Data Web App

Prerequisites

- Python 3.7 or higher
- Access to an EC2 instance to deploy the web application
- Another EC2 server that provides a REST API for student data

Setup on Your EC2 Instance

1. Install Required Packages

bash Copy

```bash
# Update package lists
sudo apt update  # For Ubuntu/Debian
# OR
sudo yum update  # For Amazon Linux/CentOS

# Install Python and pip if not already installed
sudo apt install python3 python3-pip  # For Ubuntu/Debian
# OR
sudo yum install python3 python3-pip  # For Amazon Linux/CentOS

# Create and activate a virtual environment (optional but recommended)
python3 -m venv venv
source venv/bin/activate

# Install required Python packages
pip install flask requests gunicorn
```

2. Create Application Files

Create the following directory structure:

Copy

```
/home/ec2-user/student-app/
├── app.py
└── templates/
    ├── index.html
    └── error.html
```

Copy the code from the provided files into these respective locations.

3. Configure Your Application

Edit `app.py` and update the `EC2_SERVER_URL` variable to point to your other EC2 server's API endpoint:

python Copy

```python
EC2_SERVER_URL = 'http://your-other-ec2-server-ip:port/api/students'
```

4. Run the Application

For testing:

bash Copy

```bash
python app.py
```

For production, create a systemd service file to run the application with Gunicorn:

bash Copy

```bash
sudo nano /etc/systemd/system/student-app.service
```

Add the following content:

ini Copy

```ini
[Unit]
Description=Student Data Web Application
After=network.target

[Service]
User=ec2-user
WorkingDirectory=/home/ec2-user/student-app
ExecStart=/home/ec2-user/student-app/venv/bin/gunicorn -w 4 -b 0.0.0.0:5000 app:app
Restart=always

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

Enable and start the service:

bash Copy

```bash
sudo systemctl daemon-reload
sudo systemctl enable student-app
sudo systemctl start student-app
```

5. Configure Security Groups

Make sure your EC2 instance's security group allows inbound traffic on port 5000 (or whichever port you configured).

Accessing the Application

Once deployed, you can access your application at:

Copy

```
http://your-ec2-instance-public-ip:5000
```

For a more professional setup, consider:

1. Using Nginx as a reverse proxy
2. Setting up SSL/TLS with Let's Encrypt
3. Using a domain name instead of the IP address

Troubleshooting

If you encounter issues:

1. Check the application logs:

   bash Copy

   ```
   sudo journalctl -u student-app
   ```

2. Verify connectivity to your other EC2 server:

   bash Copy

   ```
   curl http://your-other-ec2-server-url/api/students
   ```

3. Ensure both EC2 instances can communicate with each other (check security groups)