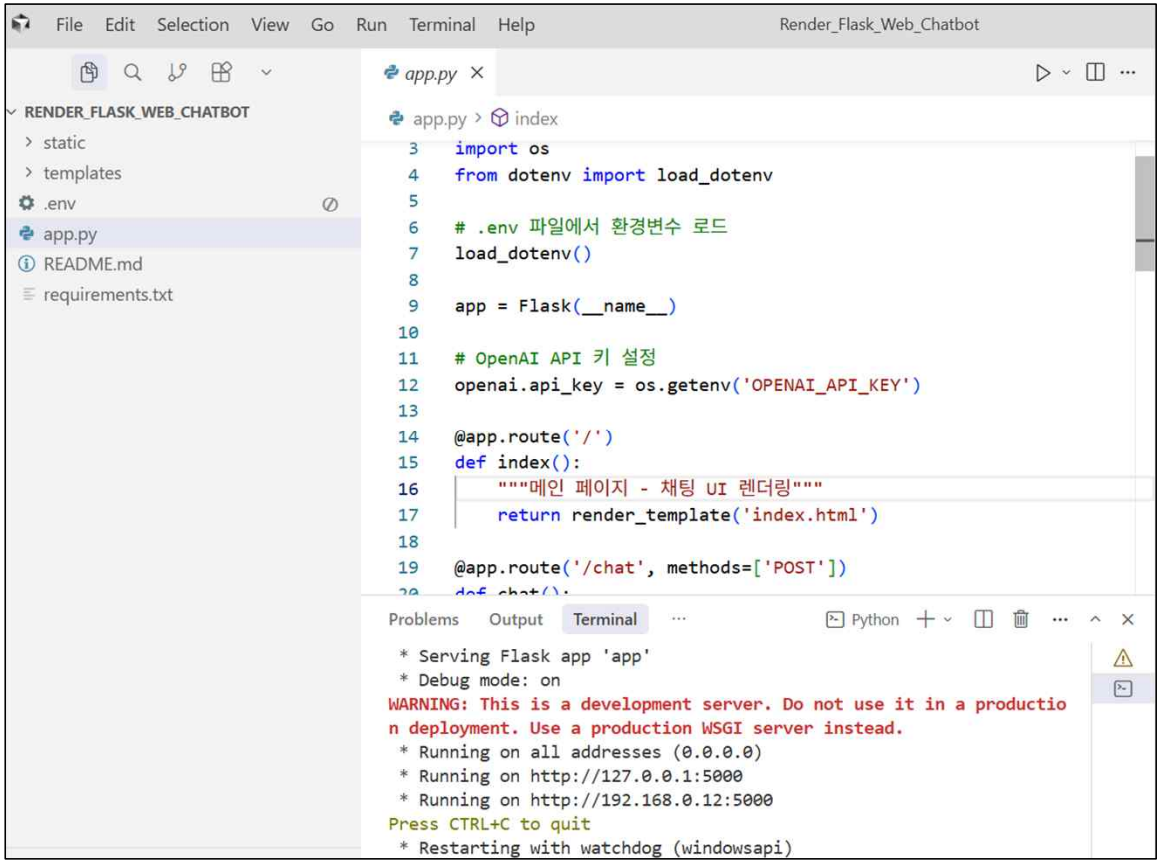
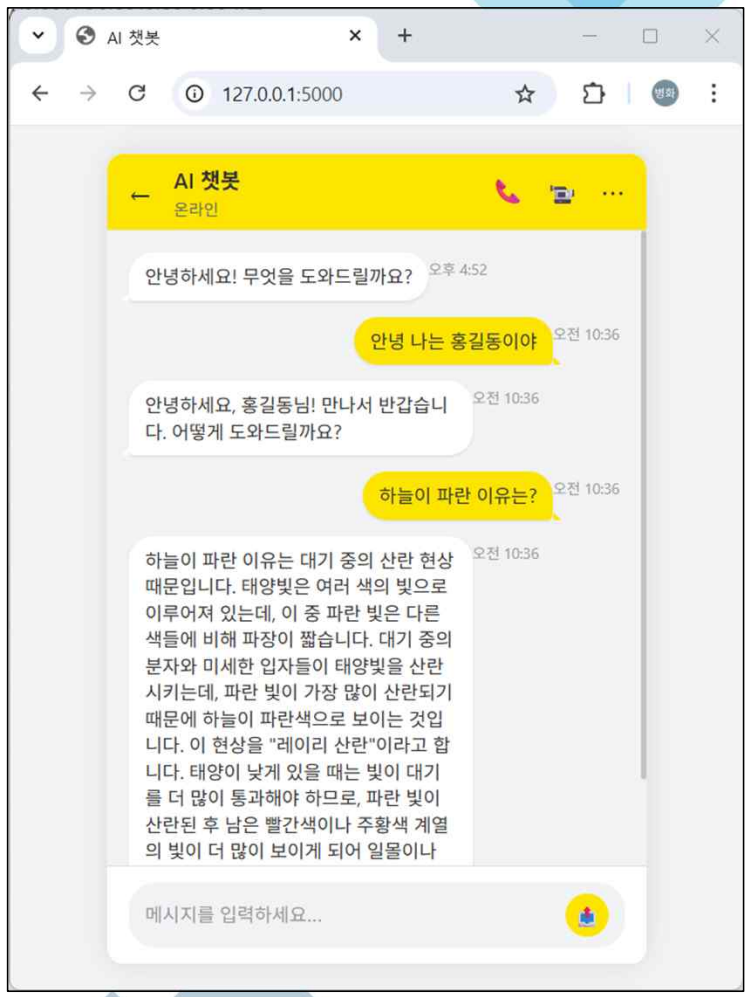
Render로 Flask Web Chatbot 배포하기

1. Flask 웹 챗봇 소스 준비

Flask 를 사용한 웹 챗봇 소스를 복사해와서 Render_Flask_Web_Chatbot 폴더로 이름을 변경하고 커서를 사용하여 동작을 확인한다





- Requirements.txt 파일에 gunicorn을 추가한다
python-dotenv는 .env를 사용하지 않으므로 필요 없으므로 삭제한다(주석 처리)

Flask==3.0.3
Openai>=1.5.0
 # python-dotenv==1.0.0
Gunicorn==22.0.0

- app.py 소스에 가서 dotenv 사용 부분(파란색 2줄)을 주석 처리하고 저장한다

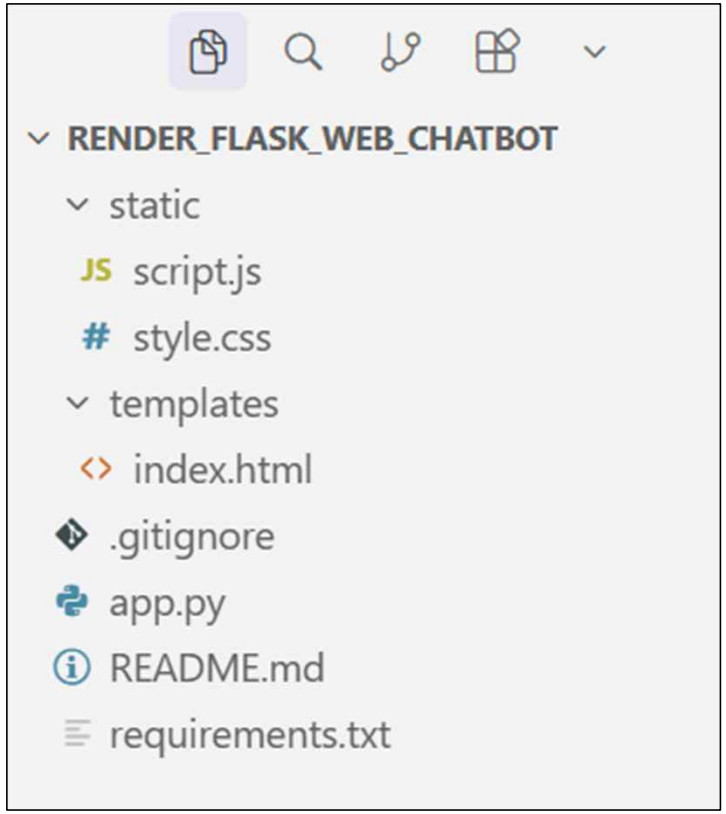
```
from flask import Flask, render_template, request, jsonify
import openai
import os
# from dotenv import load_dotenv


# .env 파일에서 환경변수 로드
# load_dotenv()
```

.gitignore 파일을 만들고 아래 내용을 넣어 커밋 시 무시해야 할 파일을 지정한다.

```
# Python
__pycache__/
*.pyc
*.pyo
*.pyd

# Virtual env
.venv/
venv/

# OS / IDE
.DS_Store
.vscode/
.idea/

# Logs
*.log
```

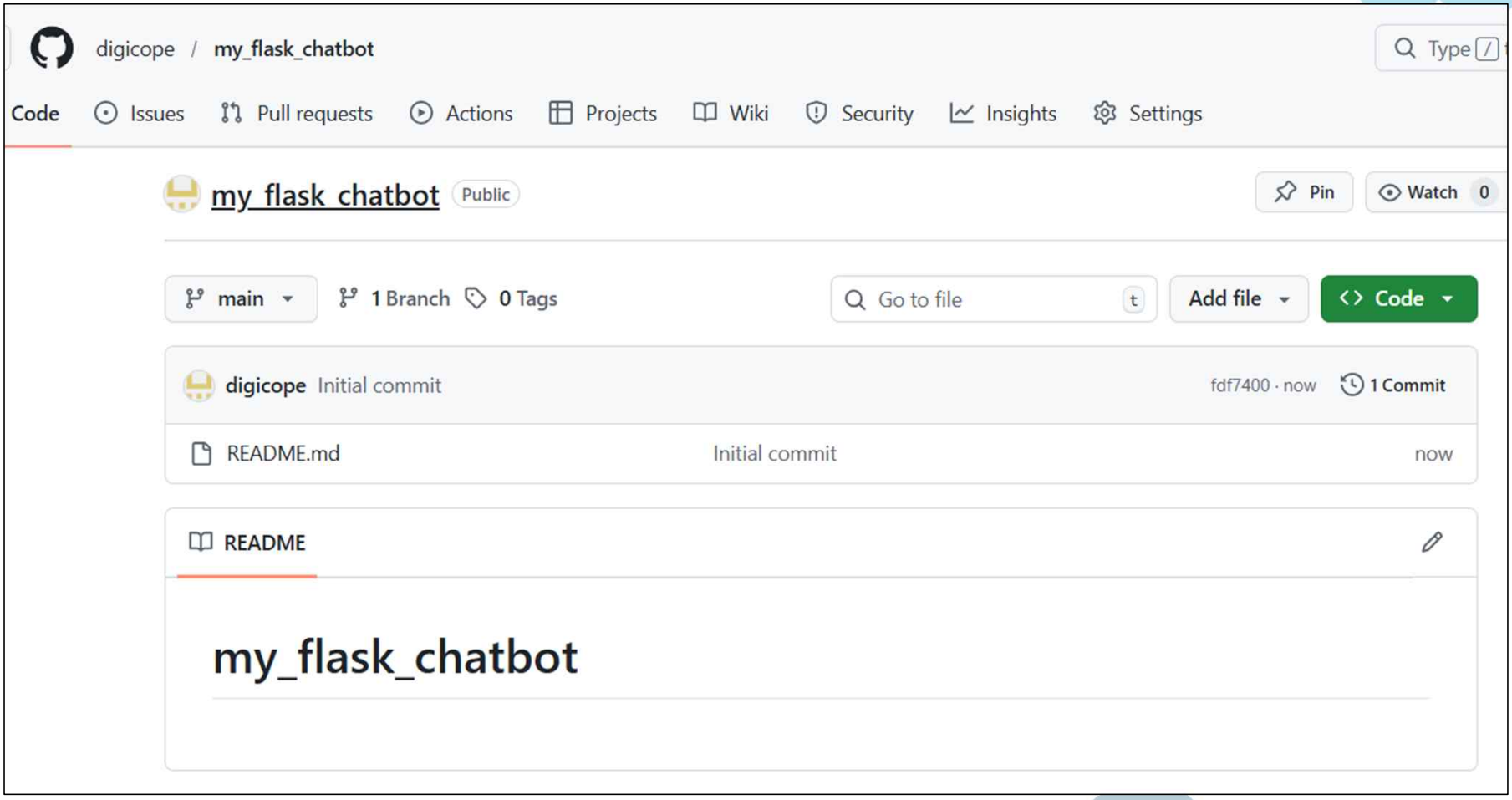
- .env 파일은 Git 저장소에 올리면 보안상 위험 하므로 삭제한다
Render에서 수동으로 환경변수에 OpenAI API 키 값을 지정해야 하므로 다른 곳에 사본을 복사해놓고 프로젝트내에서 삭제한다

변경된 프로젝트 내의 소스 파일 구조

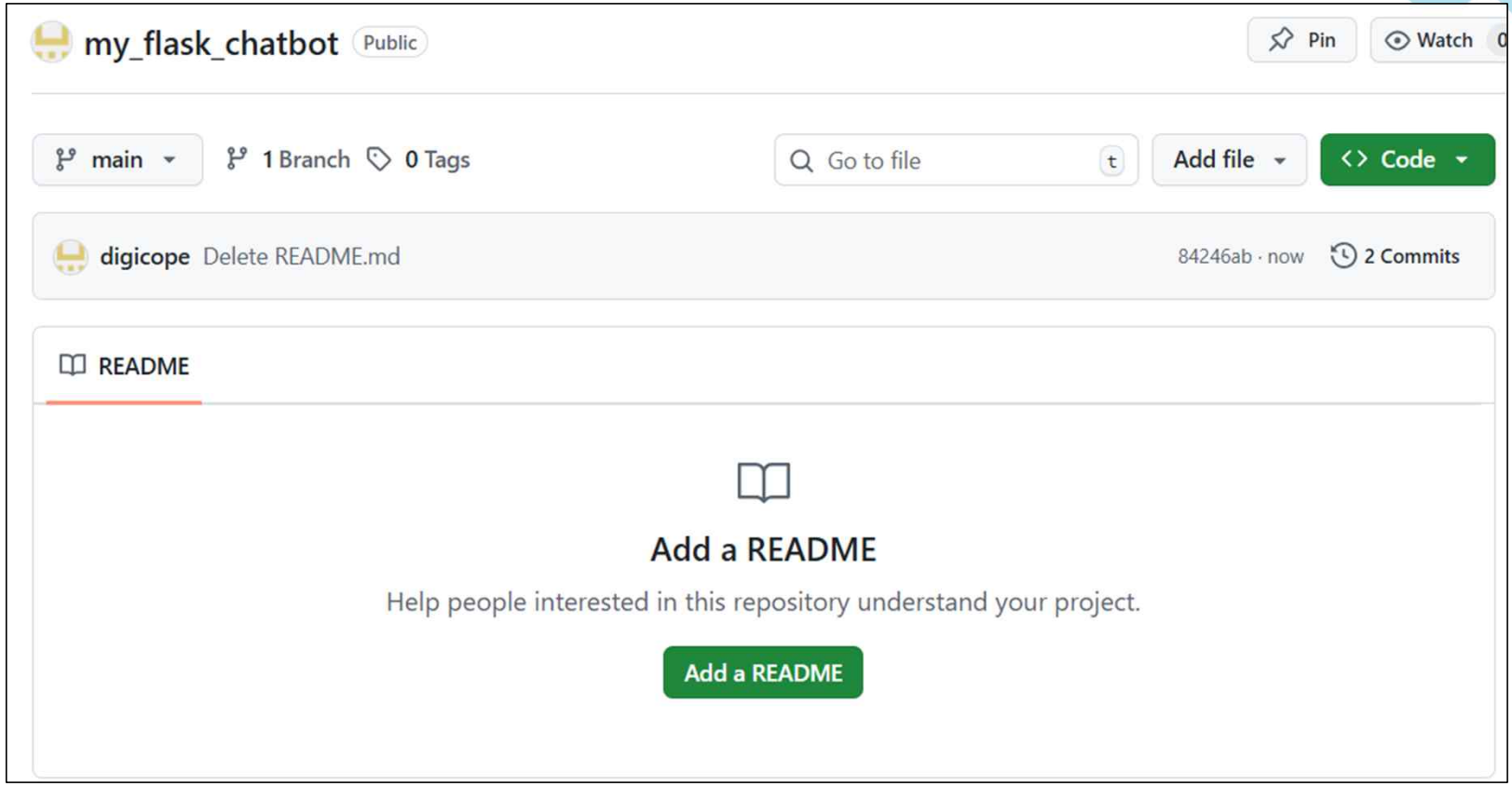


2. Git 저장소에 배포 소스 업로드

- Git에 새로운 저장소를 **my_flask_chatbot** 이름으로 생성해 놓는다

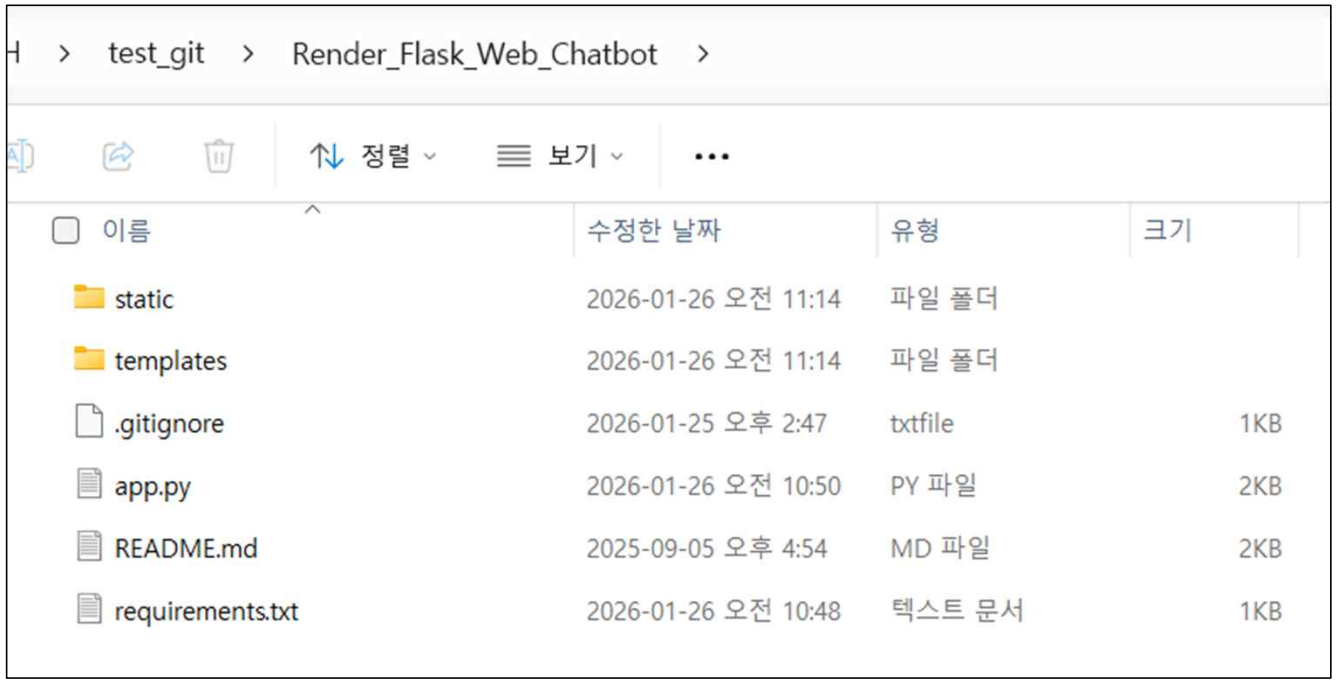


- 사용할 Github 저장소로 가서 파일을 모두 삭제해 놓는다 (README.md 파일 반드시 삭제)



- 윈도우 탐색기에서 test_git 폴더 아래 배포할 앱을 복사해 놓는다
(강사 배포 파일 사용시 : Render_Flask_Web_Chatbot.zip)

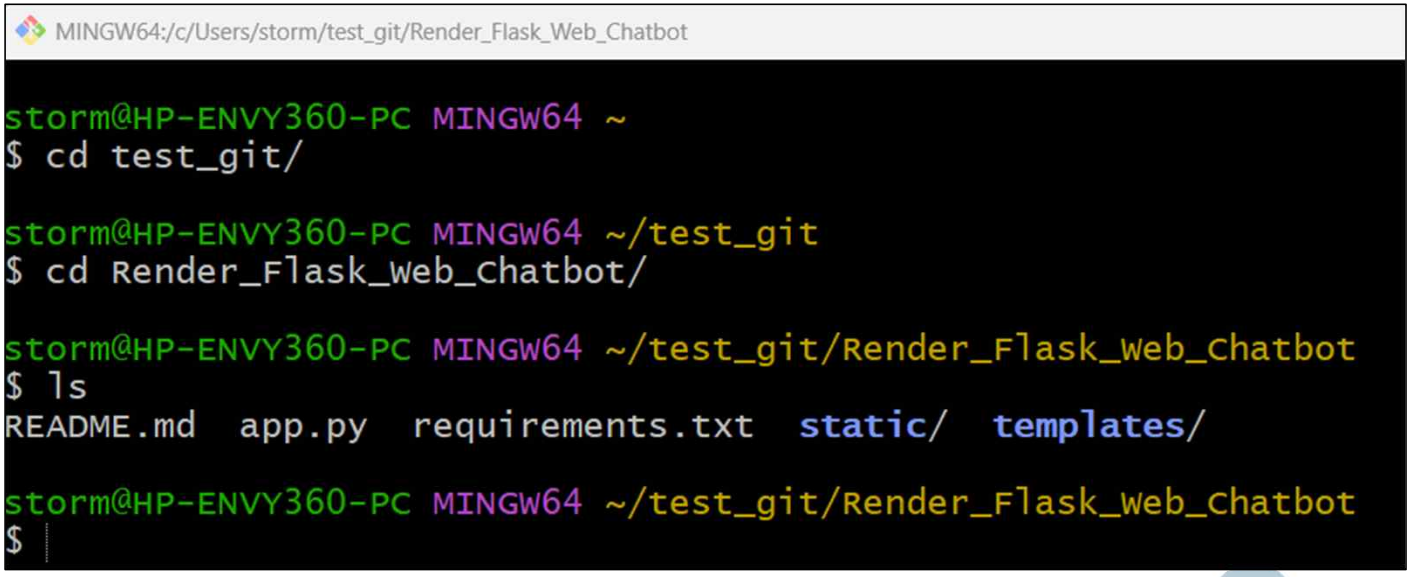
실제 경로: C:₩Users₩storm₩test_git₩Render_Flask_Web_Chatbot



- Windows에서 Git Bash를 실행하고 아래 경로로 이동한다
cd test_git

- 소스 경로로 이동한다
cd Render_Flask_Web_Chatbot

- 파일을 확인해 본다
ls

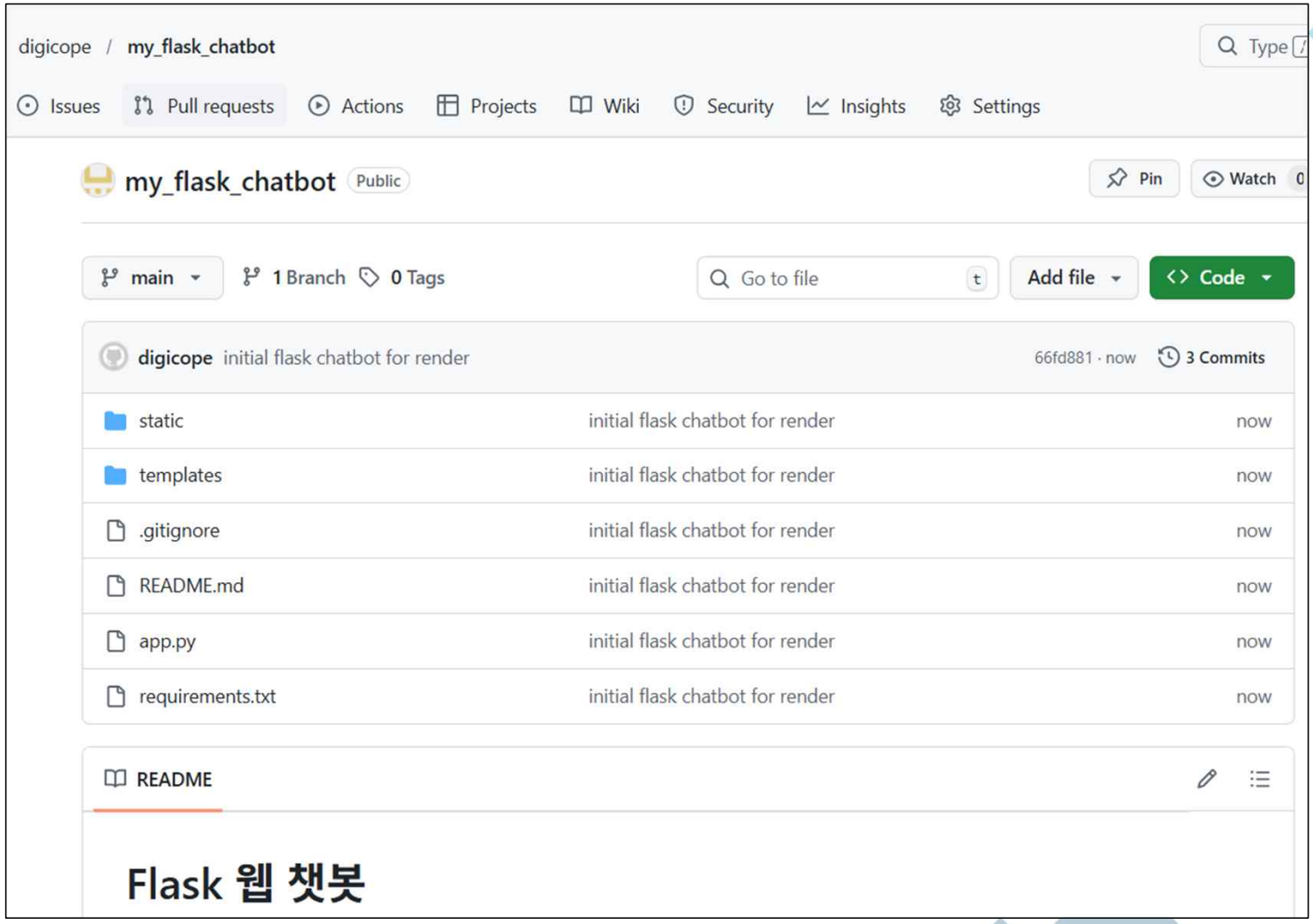


Git 저장소에 소스 파일 업로드

아래 명령을 차례로 수행한다
(**user name과 email, 저장소 경로**는 본인의 이름으로 모두 수정한 다음 실행한다)

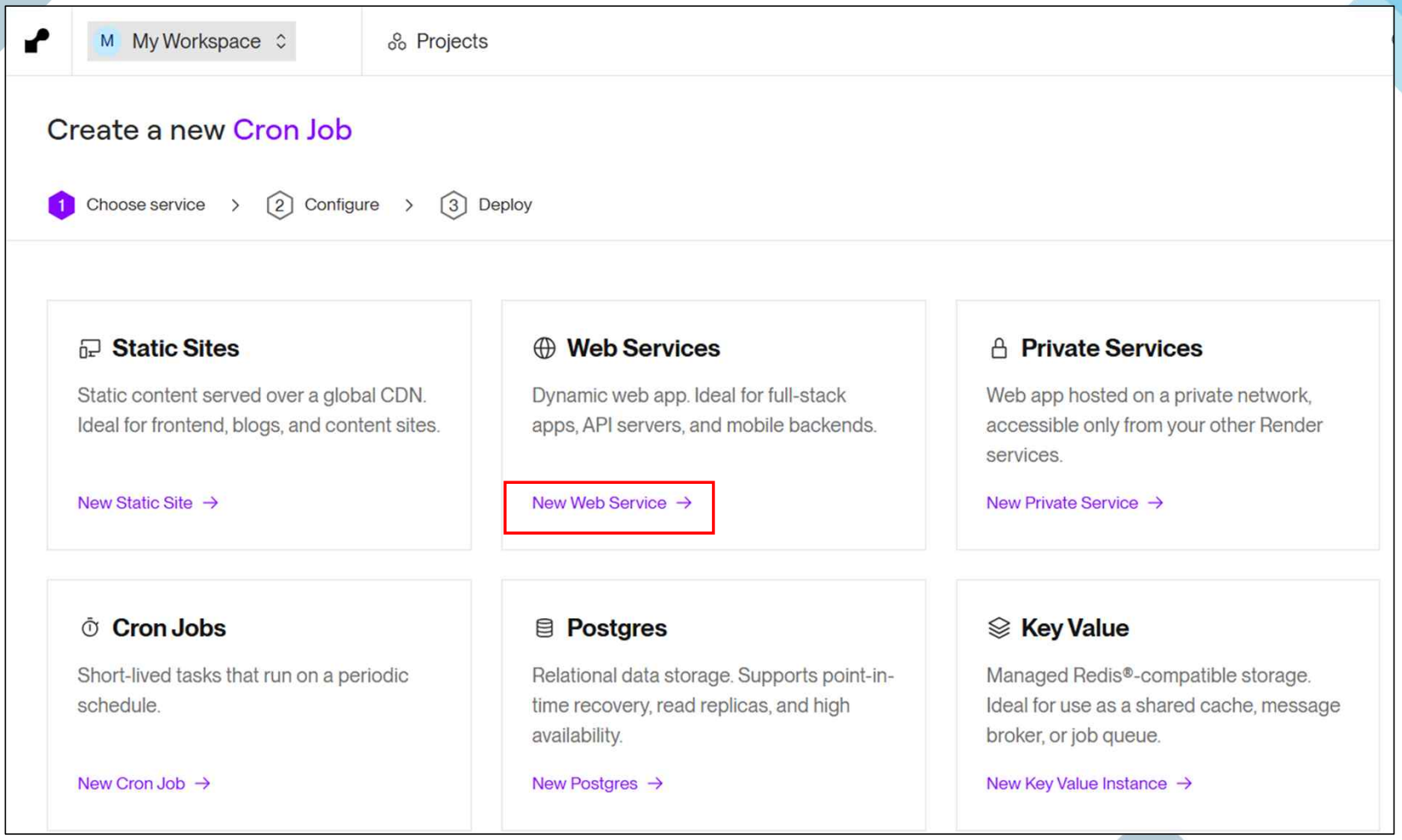
```
git init
git config --global user.name digicope
git config --global user.email digicope@aicore.co.kr

git add .
git commit -m "initial flask chatbot for render"
git branch -M main
git remote add origin
https://github.com/digicope/my_flask_chatbot.git
git pull origin main --rebase
git push -u origin main
```

Git 저장소에서 업로드 된 파일들을 확인한다



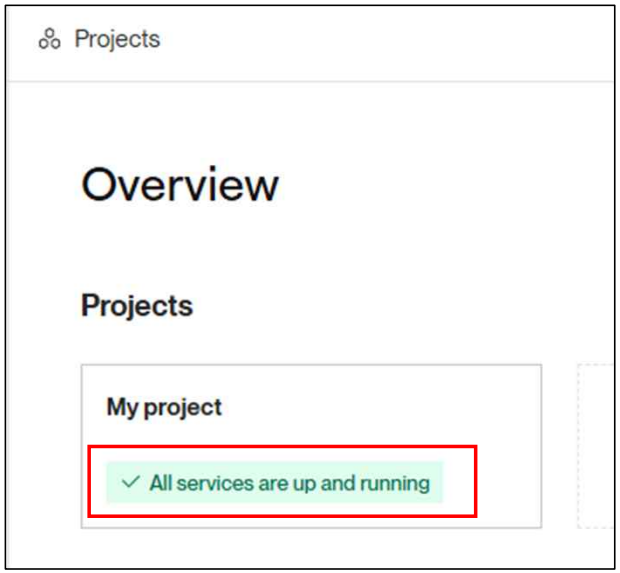
3. Render 에서 서비스 배포하기

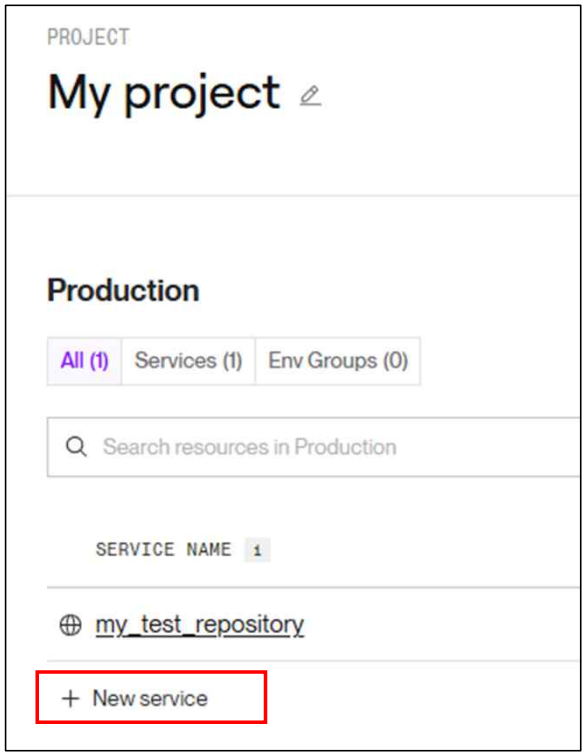
Render홈페이지 대시보드로 가서 **Web Services**의 **New Web Service** 를 누른다

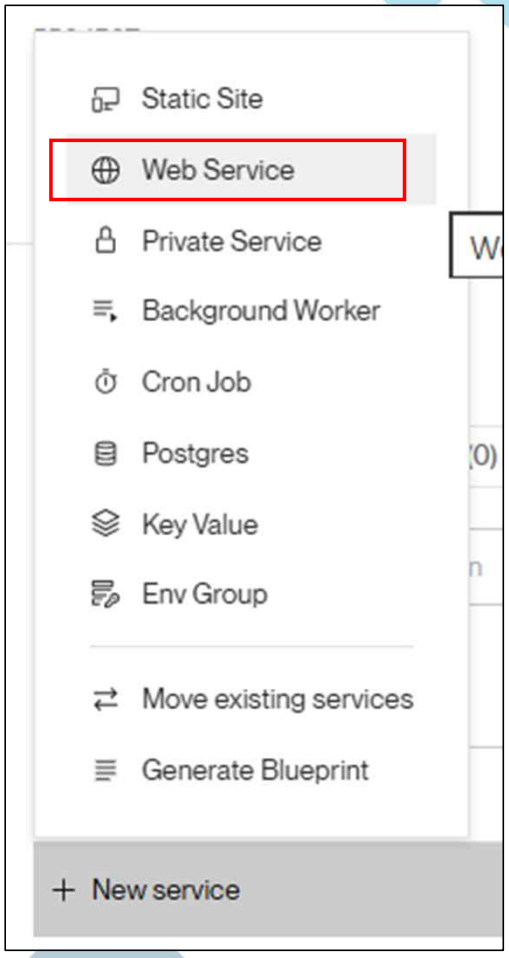


https://dashboard.render.com/

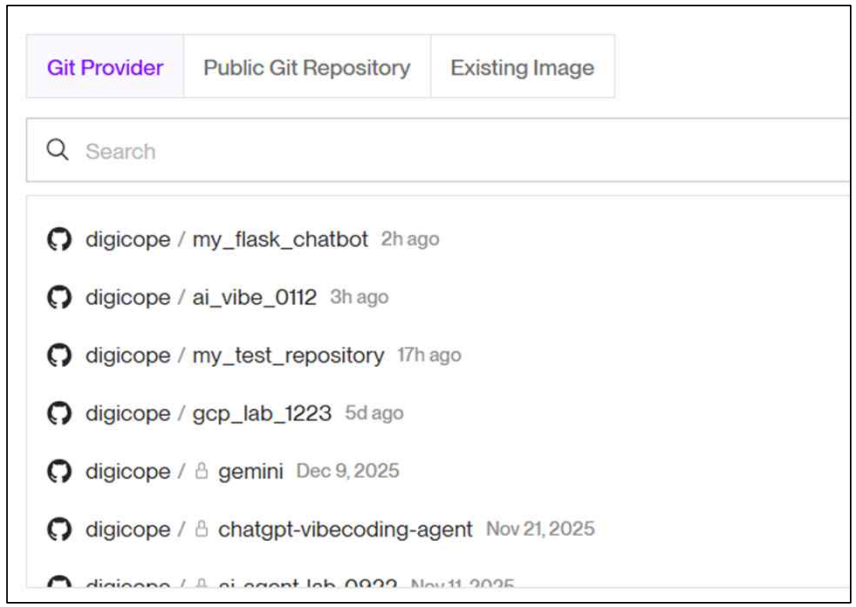
Render에서 이미 배포를 위한 프로젝트가 생성 되어 있을 경우에는 **Projects**에 있는
My project를 아래를 클릭하고 하단의 **New service**를 클릭한다음 **[Web Service]**를 클릭한다







New Web Service에서 배포에 사용할 저장소 my_flask_chatbot을 선택해준다



저장소를 선택 후 다음 항목을 설정한다.
대부분 기본 값으로 사용하면 된다

- **Name** : 서비스 이름 (기본값인 **저장소 이름**을 사용한다)
- **Environment** : **Python 3**
- **Region** : 기본값 사용 (**Oregon (US West)**)
- **Branch** : **main** (또는 master)
- **Build Command** : **pip install -r requirements.txt**
- **Start Command** : **gunicorn app:app** (파일명이 app.py이고 Flask 객체가 app인 경우)

Source Code

digicope / my_flask_chatbot • 2h ago Edit

Name

A unique name for your web service.

my_flask_chatbot

Project Optional

Add this web service to a project once it's created.

My project / Production

Language

Choose the runtime environment for this service.

Python 3

Branch

The Git branch to build and deploy.

main

Region

Your services in the same region can communicate over a private network. You currently have services running in **Oregon**.

Oregon (US West) 1 existing service

Deploy in a new region +

Root Directory Optional

If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a monorepo.

e.g. src

Build Command

Render runs this command to build your app before each deploy.

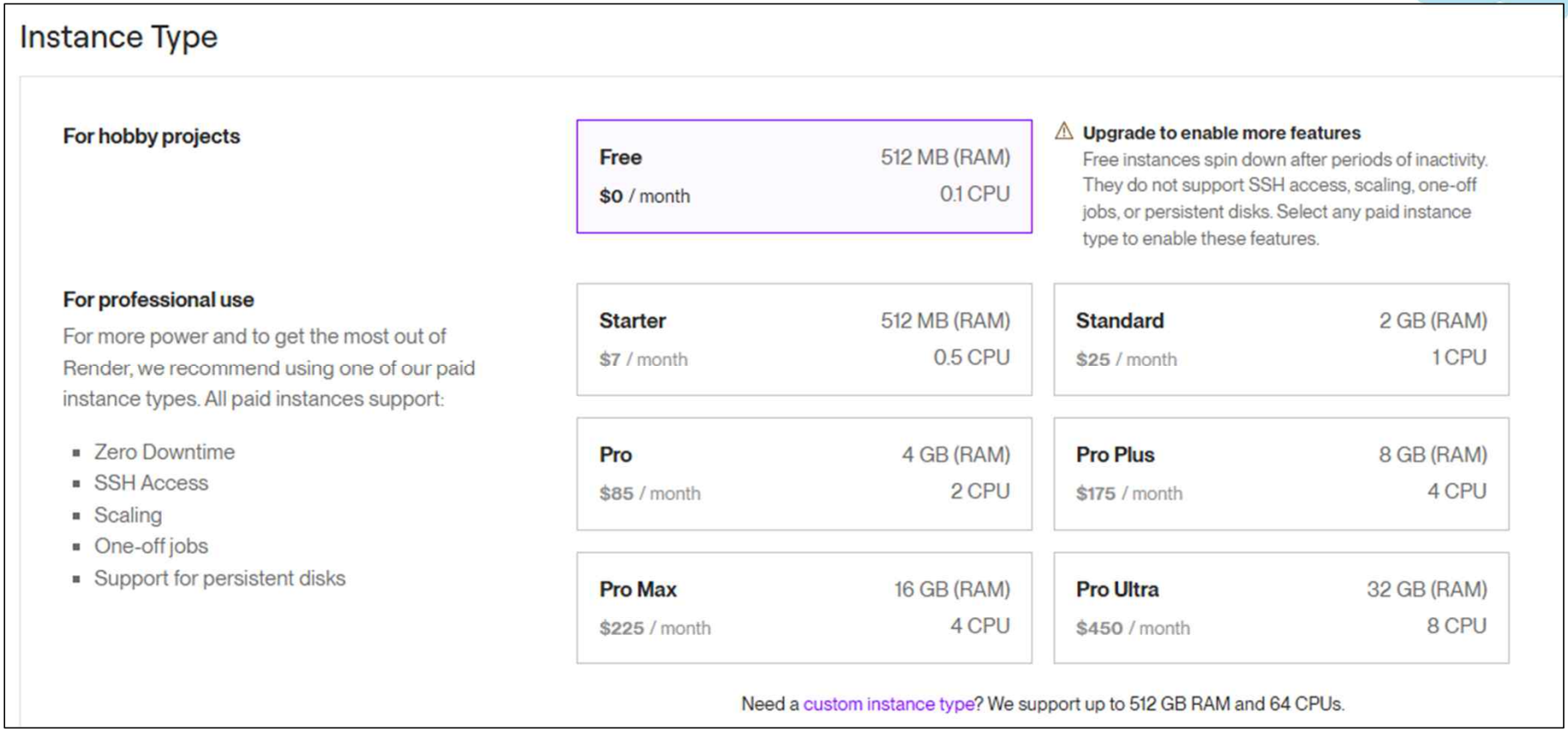
```
$ pip install -r requirements.txt
```

Start Command

Render runs this command to start your app with each deploy.

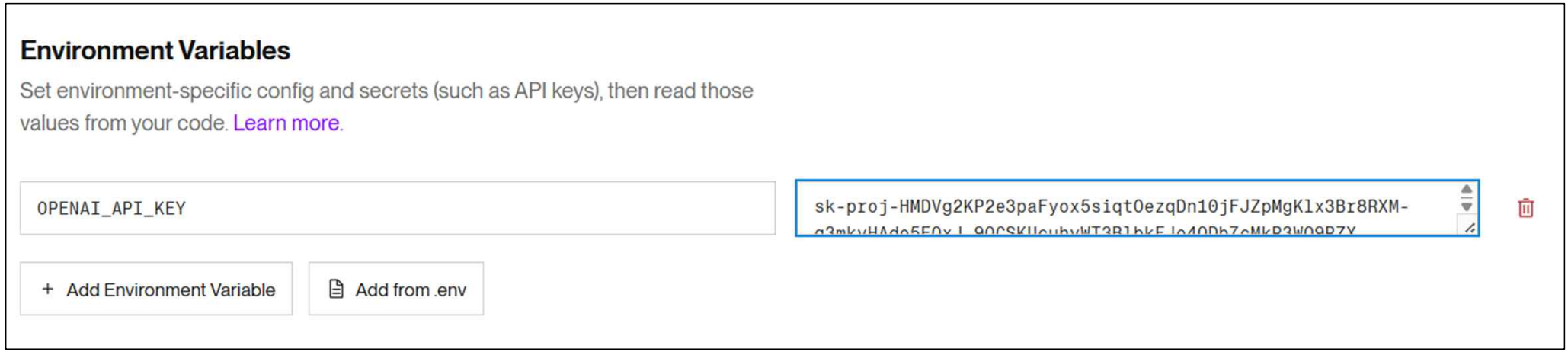
```
$ gunicorn app:app
```

Instance Type을 Free 로 선택한다

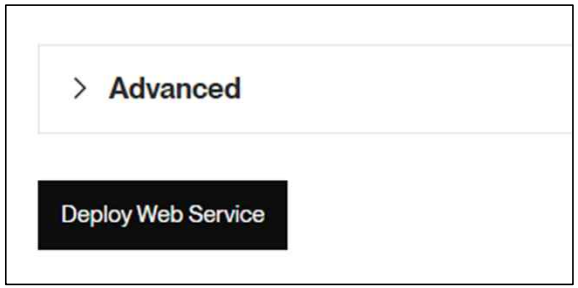


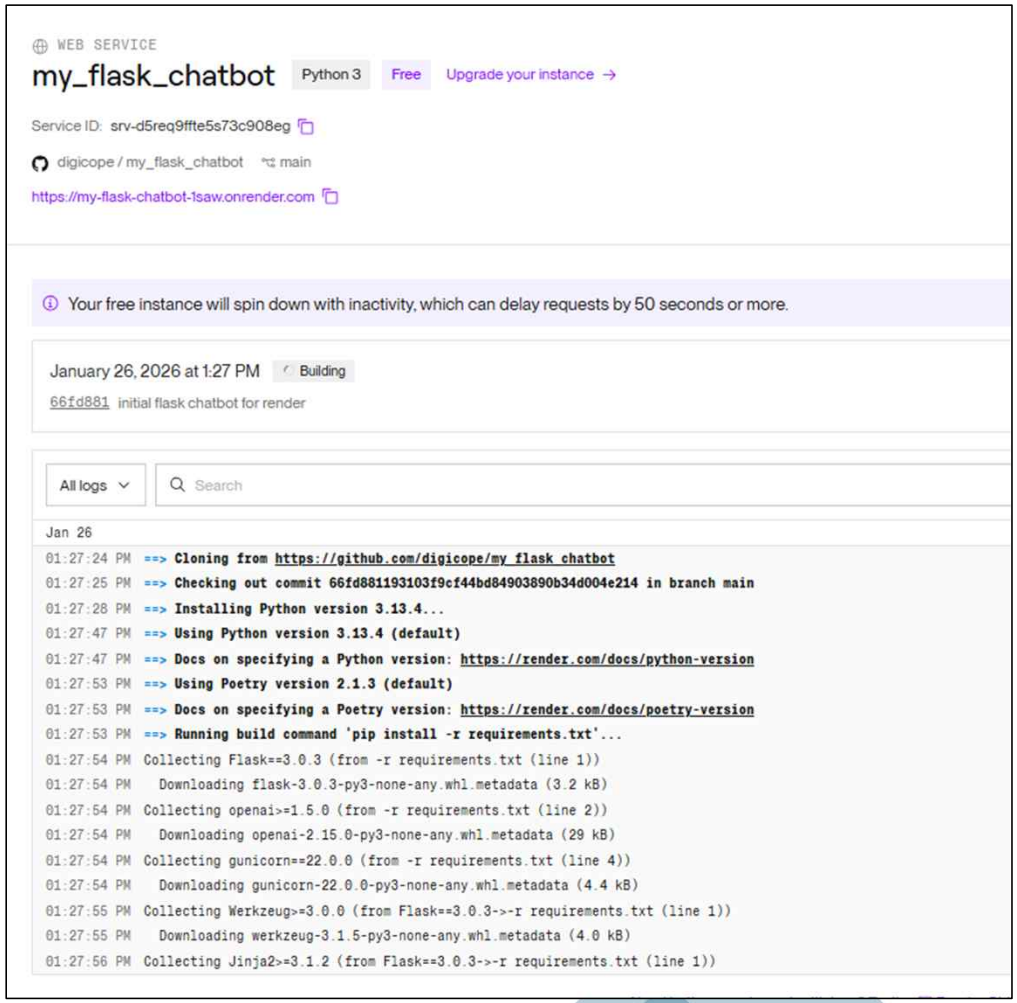
Environment Variables 에 .env에 있는 환경변수를 입력한다

NAME_OF_VARIABLE : **OPENAI_API_KEY**
Value : **sk-projxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**



Web Service 배포를 위한 설정이 완료되면 좌측 하단의 [Deploy Web Service]를 누르면 아래와 같이 배포가 진행 된다

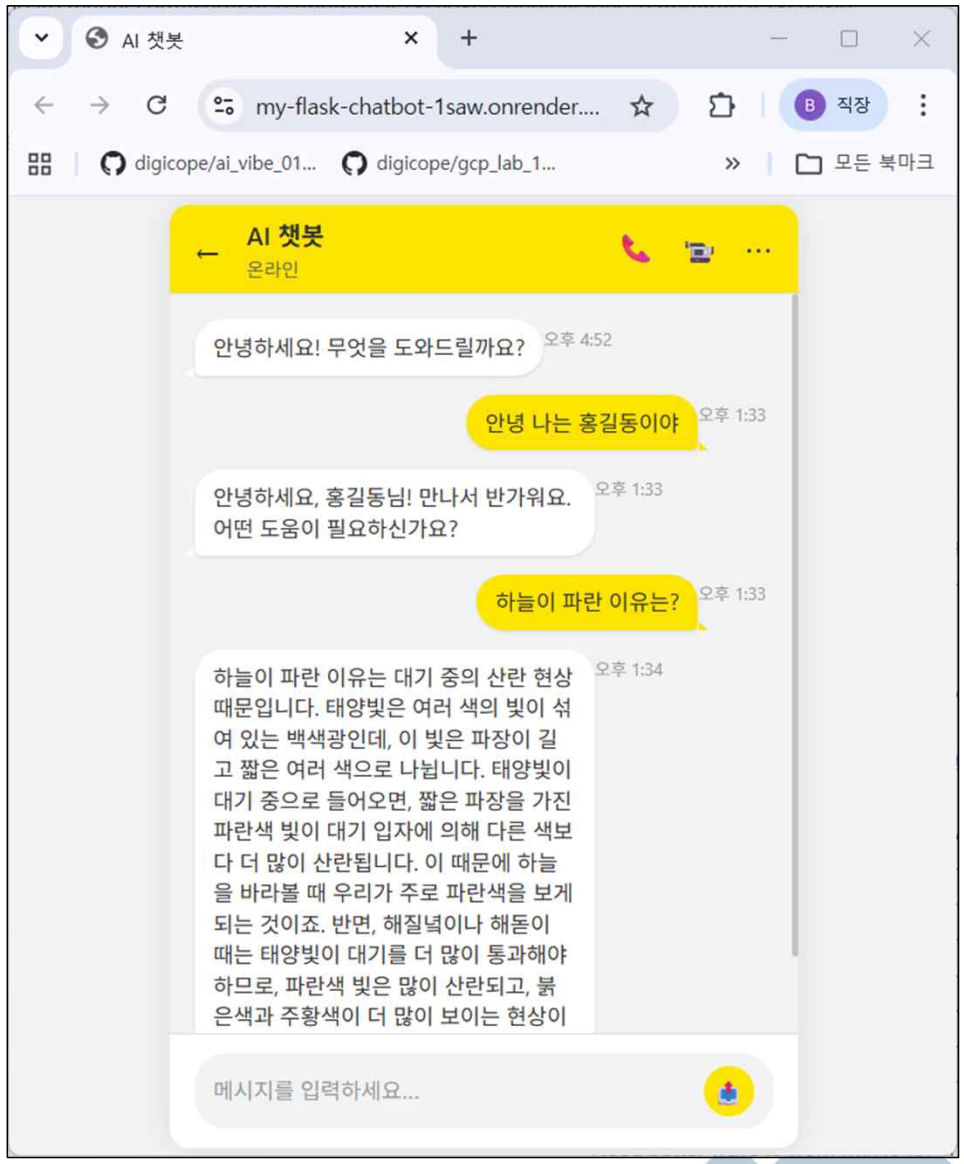




잠시 기다리면 아래와 같이 접속 가능한 URL이 보여진다
URL을 클릭하면 배포된 챗봇 웹 서버로 접속된다

```
01:33:19 PM          ==> Your service is live 🎉
01:33:19 PM          ==>
01:33:19 PM          ==> ///////////////////////////////////////////////////////////
01:33:20 PM          ==>
01:33:20 PM          ==> Available at your primary URL https://my-flask-chatbot-1saw.onrender.com
01:33:20 PM          ==>
01:33:20 PM          ==> ///////////////////////////////////////////////////////////
01:33:20 PM  [wtvmd] 127.0.0.1 - - [26/Jan/2026:04:33:20 +0000] "GET / HTTP/1.1" 200 1778 "-" "Go-http-client/2.0"
```

배포 서비스 접속 실행 화면



4. Render 에서 배포 서비스 업데이트 및 삭제

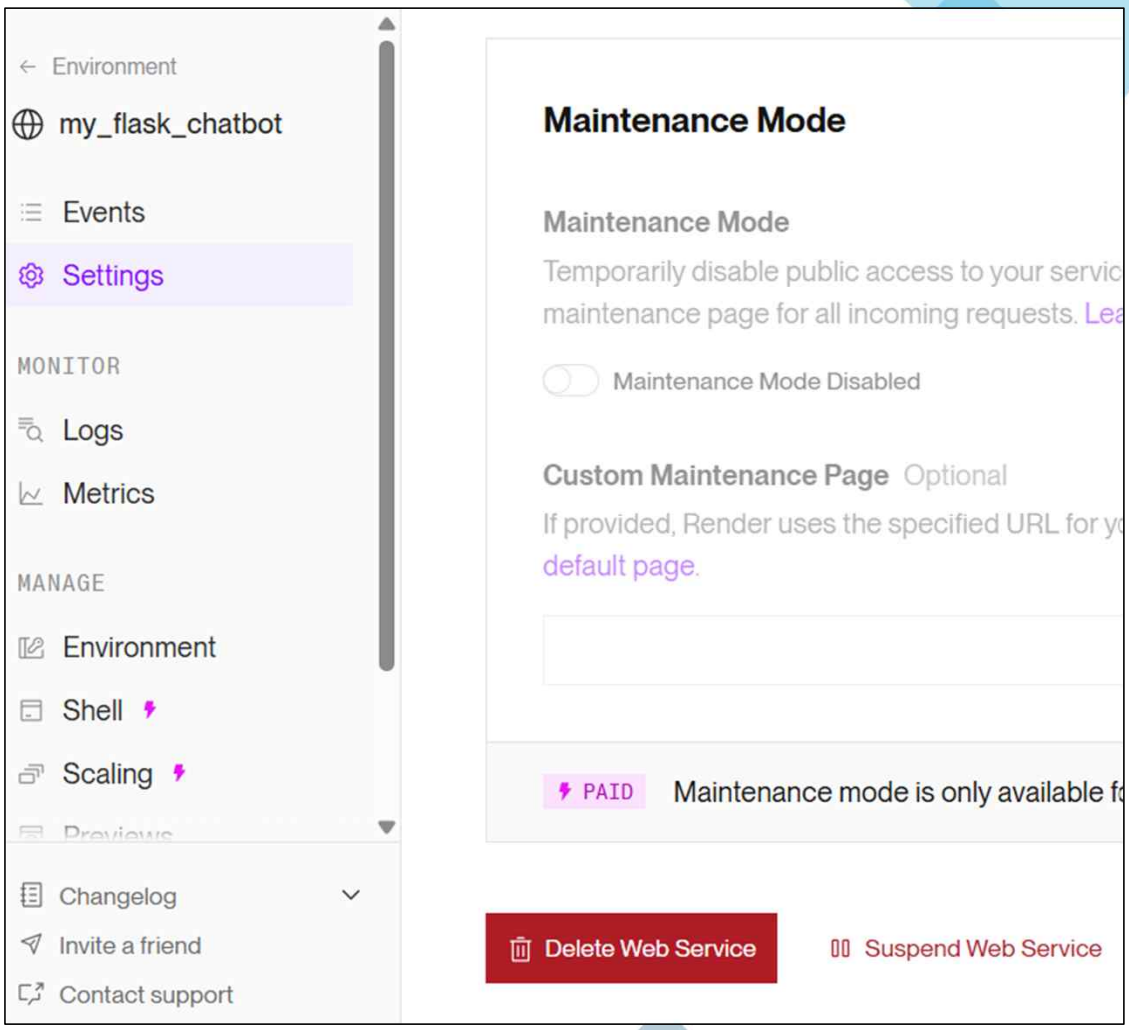
배포된 소스 업데이트 하기

- app.py 소스를 수정하고 Git Bash 아래 명령을 수행하면 자동 배포 업데이트 된다

git add .
git commit -m "update flask chatbot for render"
git push origin main

배포된 웹 서비스 삭제하기

render.com → Dashboard
→ Project : My project
→ Production : my_flask_chatbot
→ 좌측 메뉴의 Settings 클릭
→ 맨 아래의 [Delete Web Service] 를 클릭한다



Git hub의 모든 파일 삭제하기 (삭제후 복구 불가능)

- Git Bash에서 작업 디렉토리에서 아래 명령을 수행한다

git checkout --orphan main
git rm -rf .

echo "# clean repo" > README.md
git add README.md
git commit -m "initial commit"
git push -f origin main

(업로드 된 파일이 복잡해서 삭제가 잘 안될 때는 저장소 자체를 삭제하고 다시 만든다)

감사합니다