

Cloud Deployment – The Heroku Platform

Samkeun Kim <skim@hknu.ac.kr>

<http://cyber.hknu.ac.kr/>

Cloud Deployment

Cloud hosting:

- EC2 service from Amazon Web Service (AWS)
- Container Platform (Docker)
- Google App Engine
- ...
- Platform as a Service (PaaS) – **Heroku** Platform

PaaS 모델:

서비스 제공 업체 => 애플리케이션을 실행할 수 있는 완전히 관리되는 플랫폼 제공

애플리케이션 개발자 => 서비스 제공 업체가 유지 관리하는 서버에 애플리케이션 코드 업로드
(업로드한 후 보통 몇 초 내에 자동으로 사용 가능하게 된다)

The Heroku Platform

- Heroku는 2007년부터 사업을 시작한 최초의 PaaS 제공 업체 중 하나
- Heroku 플랫폼은 매우 유연하며 Python을 포함한 Java, Node.js 등의 다양한 프로그래밍 언어들을 지원
- 애플리케이션을 Heroku에 배포하기 위해 Git을 사용하여 애플리케이션을 Heroku의 특수 Git 서버에 푸시하면 애플리케이션의 설치, 업그레이드, 설정 및 배포가 자동으로 트리거 됨
- Heroku는 dynos라는 컴퓨팅 단위를 사용하여 서비스 사용량과 요금을 측정
- dyno의 가장 일반적인 유형은 웹 서버 인스턴스를 나타내는 **web dyno**임
- 애플리케이션은 각각 웹 애플리케이션의 인스턴스를 실행하는 더 많은 웹 dyno를 배포하여 요청 처리 용량을 늘릴 수 있음
- 다른 유형의 dyno로는 백그라운드 작업 또는 기타 지원 작업을 수행하는 데 사용되는 **worker dyno**가 있음

Preparing the Application

Heroku => Git repository에 애플리케이션을 호스팅해야 함

Git 설치 : <https://git-scm.com/downloads>

Git 사용법: [제 17.1강 Git – 버전관리시스템.pdf](#)

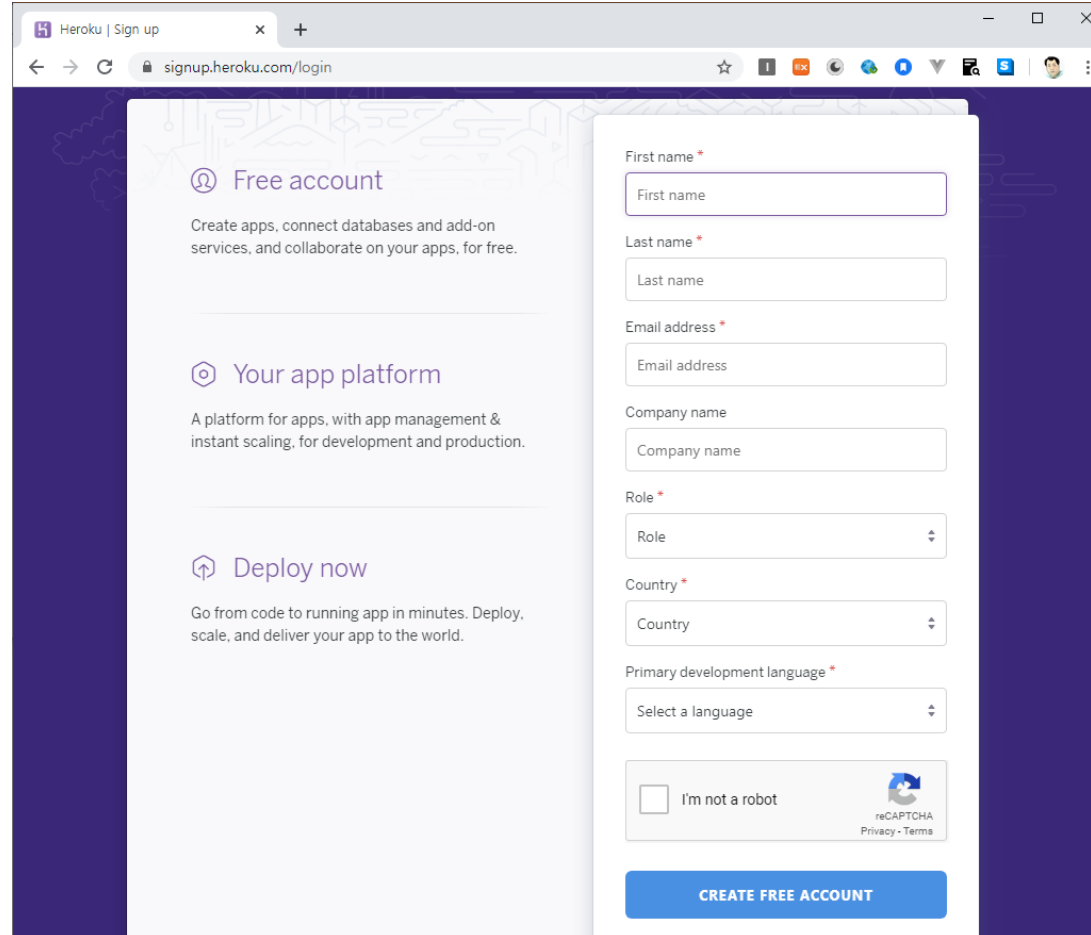
Heroku는 1개의 이메일당 5개의 앱까지 무료로 배포할 수 있도록 해준다.

⇒ 5개 이상의 앱 배포가 필요하면 또 다른 이메일 계정을 생성하면 5개의 앱을 더 배포할 수 있다...

Creating a Heroku account

Heroku에 Account 만들기

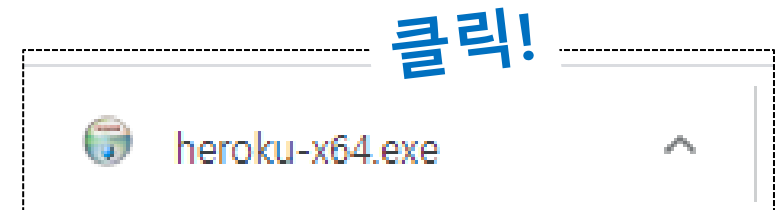
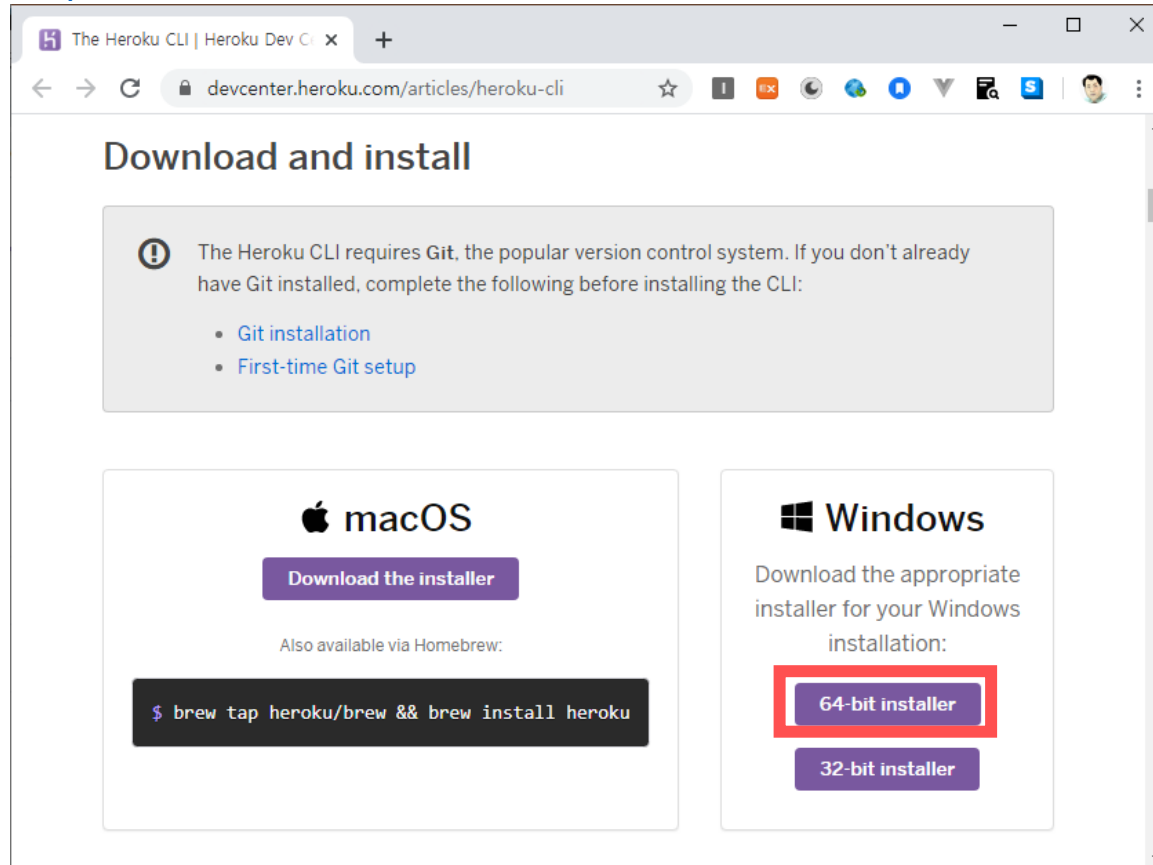
- <https://id.heroku.com/login>

A screenshot of the Heroku sign-up page in a web browser. The browser's address bar shows 'signup.heroku.com/login'. The page has a dark purple header and a light gray main content area. On the left, there are three sections: 'Free account' with a description 'Create apps, connect databases and add-on services, and collaborate on your apps, for free.', 'Your app platform' with a description 'A platform for apps, with app management & instant scaling, for development and production.', and 'Deploy now' with a description 'Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.' On the right, there is a sign-up form with fields for 'First name', 'Last name', 'Email address', 'Company name', 'Role' (a dropdown menu), 'Country' (a dropdown menu), and 'Primary development language' (a dropdown menu). Below these fields is a reCAPTCHA checkbox labeled 'I'm not a robot' and a link to 'Privacy - Terms'. At the bottom of the form is a blue button labeled 'CREATE FREE ACCOUNT'.

Installing the Heroku CLI

Heroku 서비스를 사용하기 위해서는 Heroku CLI를 설치해야

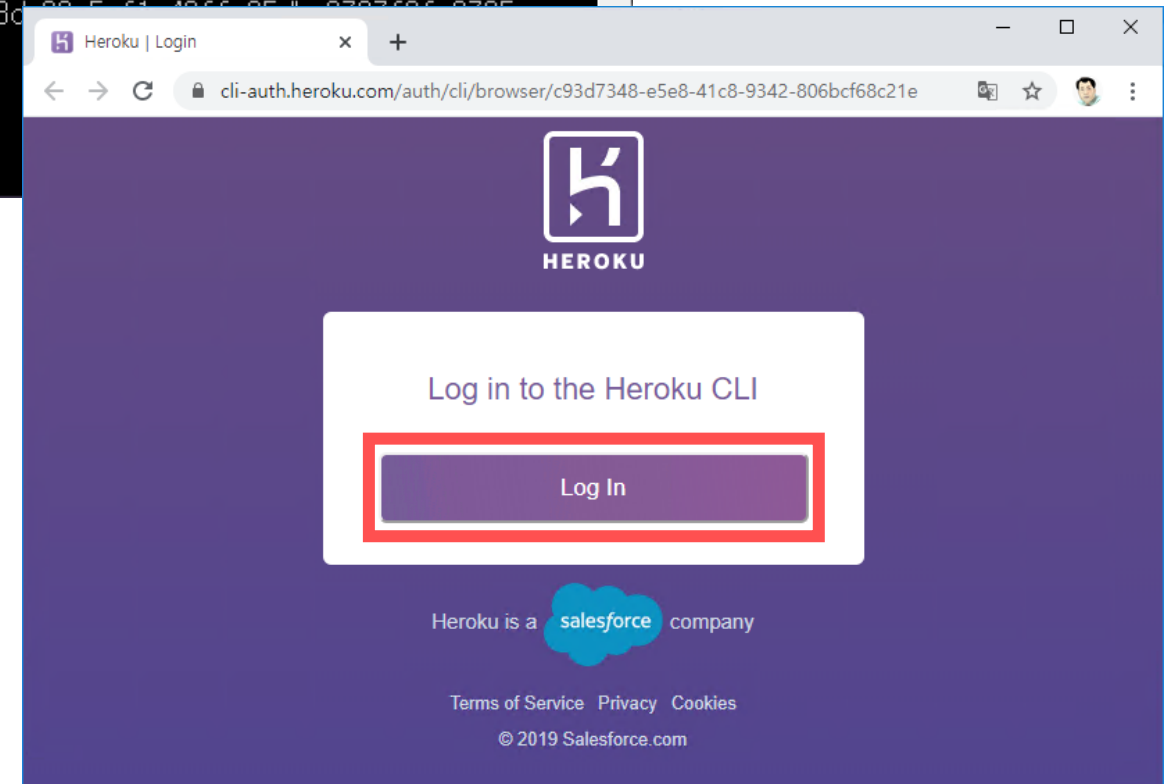
- <https://devcenter.heroku.com/articles/heroku-cli>



Anaconda prompt:

\$ heroku login

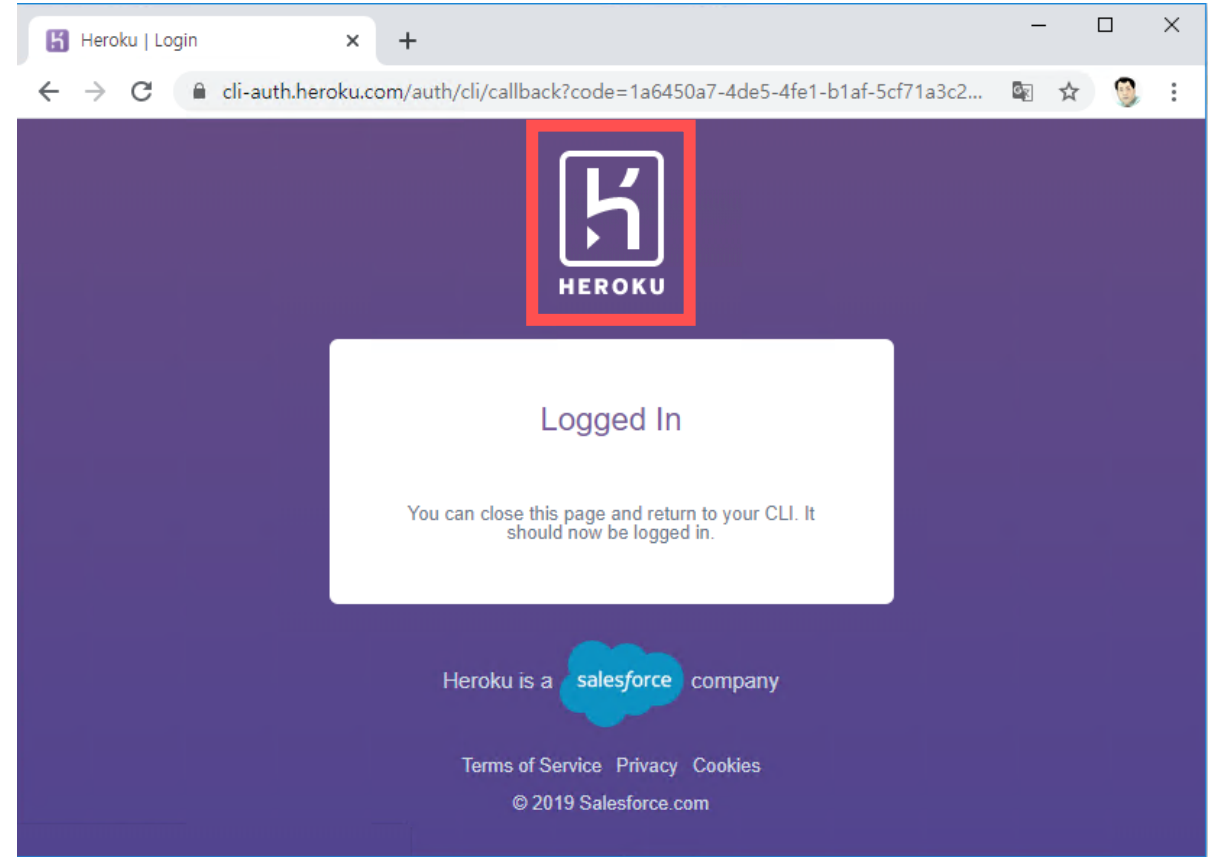
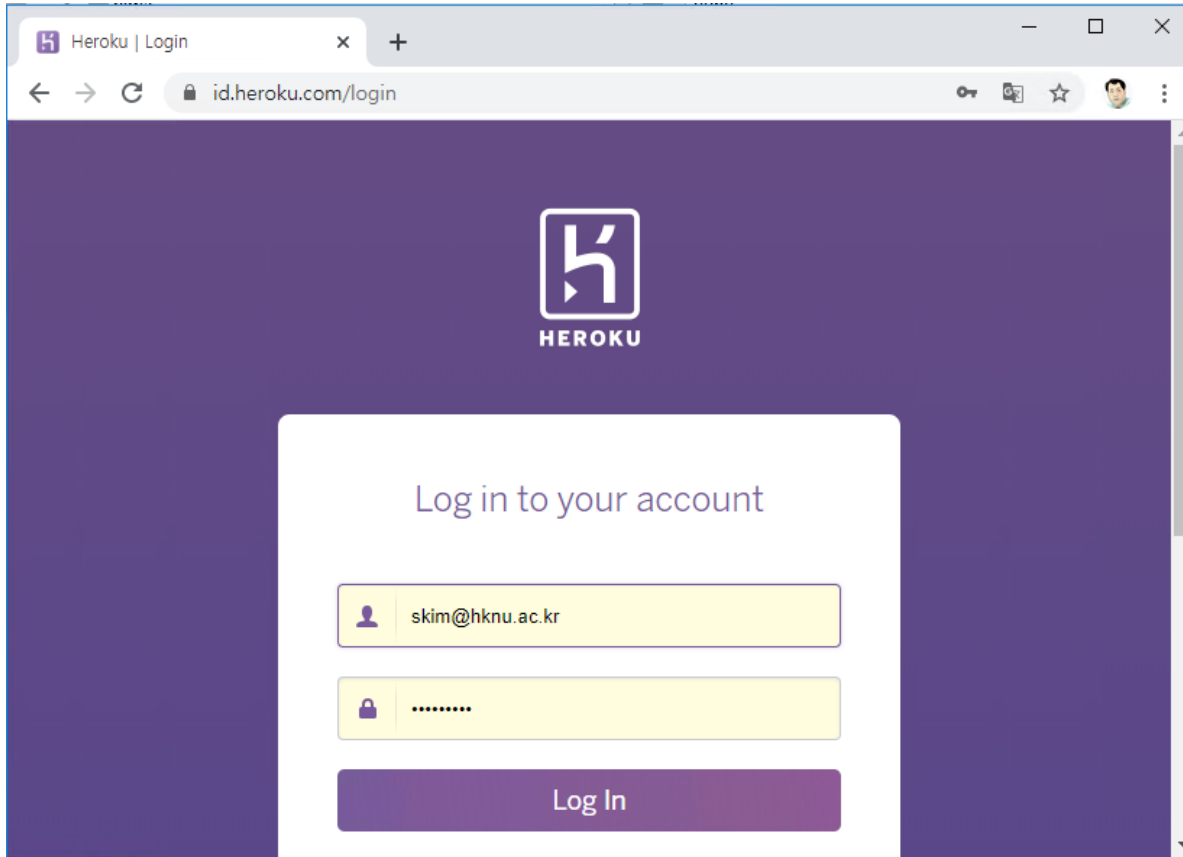
```
Anaconda Prompt (Anaconda3)
(base) C:\Users\User>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/a473d...
Logging in... done
Logged in as skim@hknu.ac.kr
(base) C:\Users\User>_
```



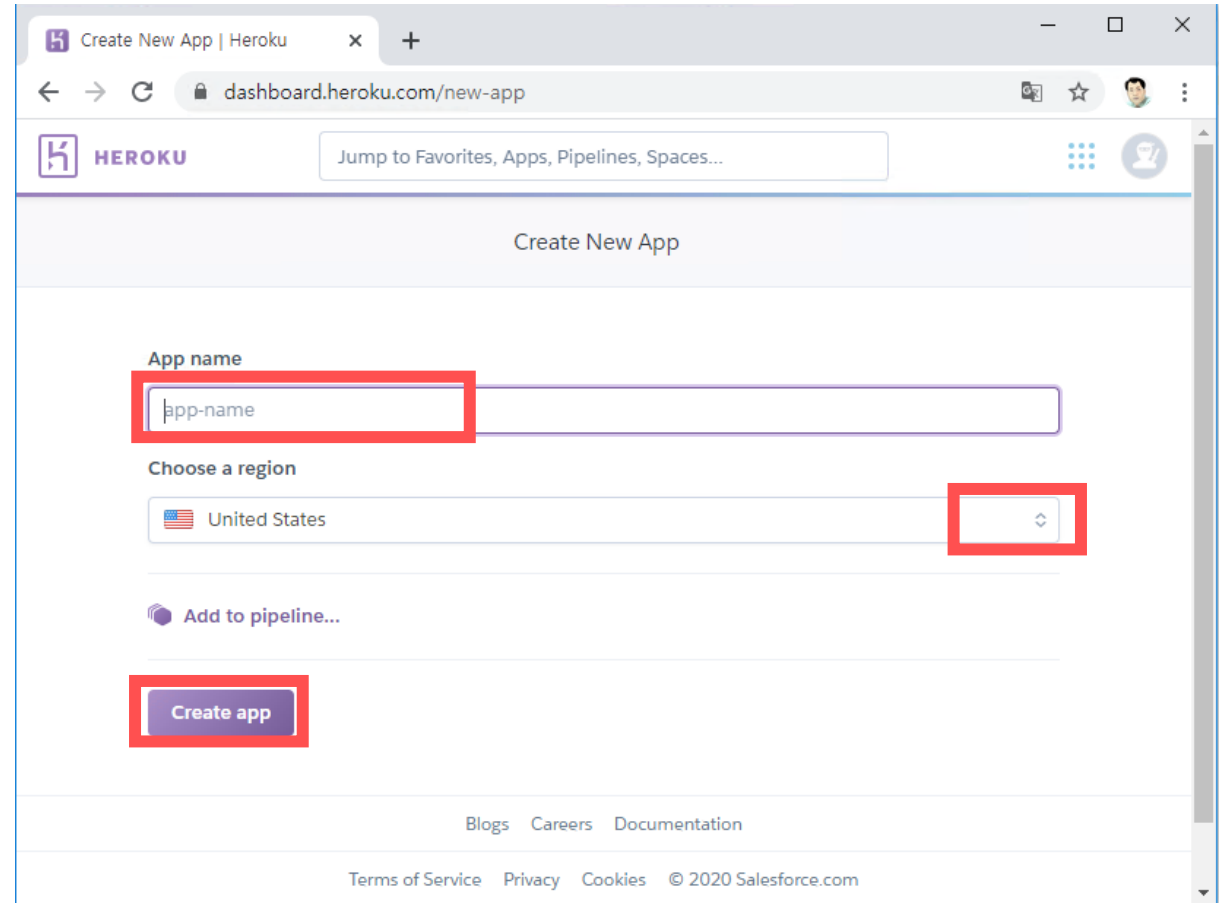
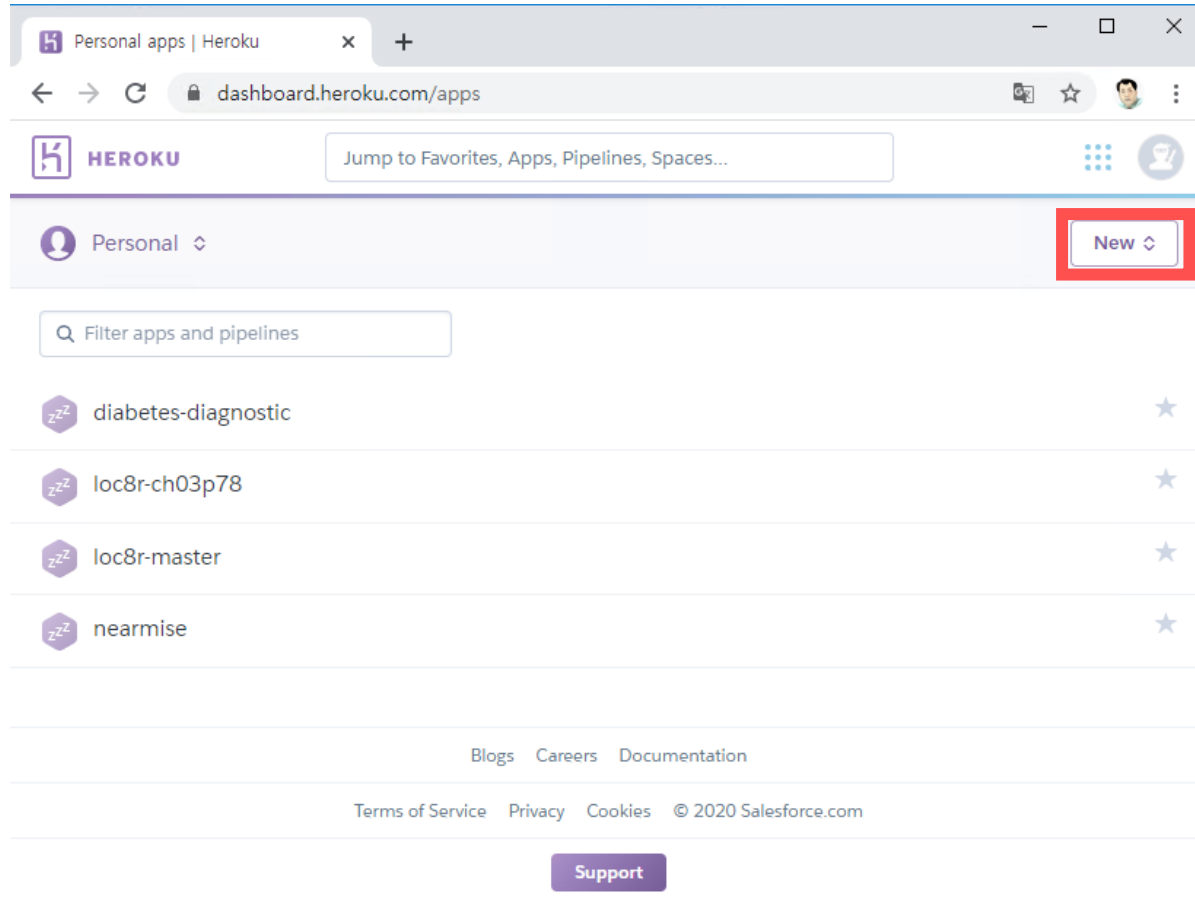
또는

<https://id.heroku.com/login>

Installing the Heroku CLI



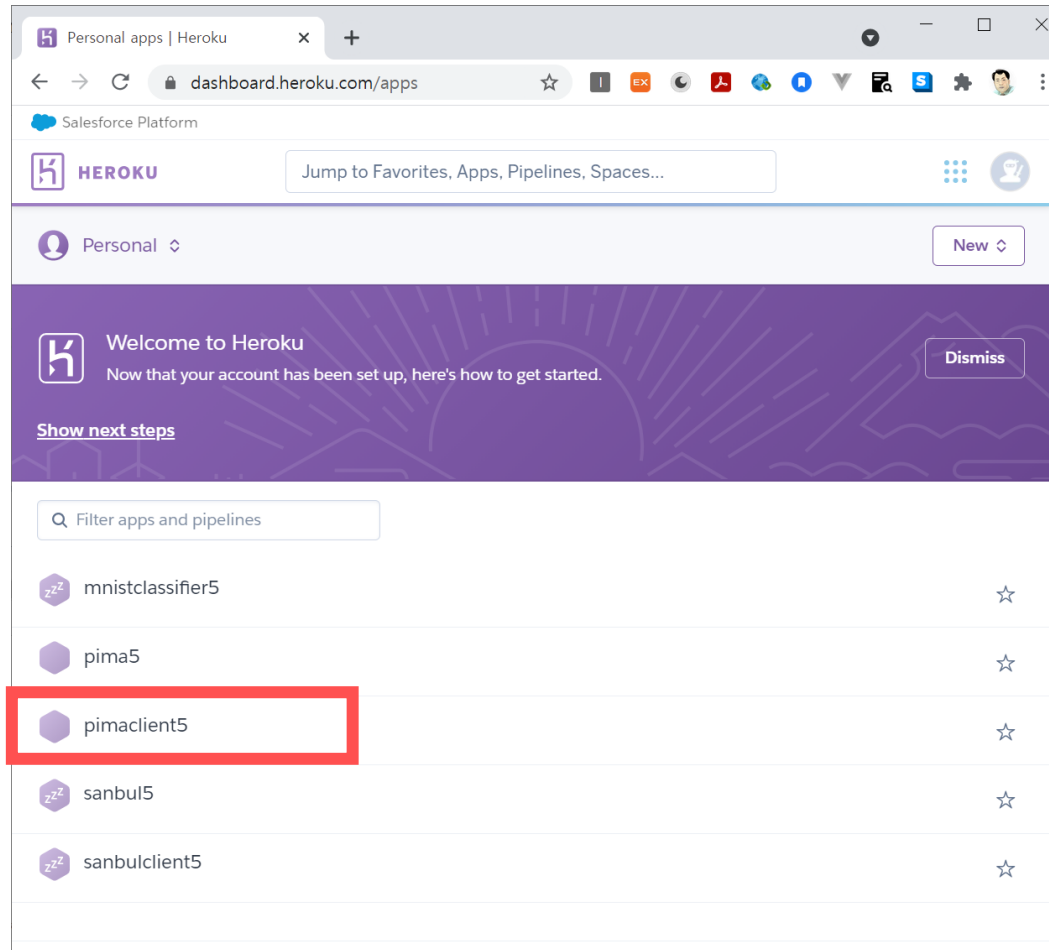
Creating an application



Flask Web App을 Heroku에 호스팅하기:

> heroku login

앱 생성 =>



Adding a top-level requirements file

Heroku에서 애플리케이션을 실행하기 위해서는 적절한 환경과 의존성을 셋업해야 한다:

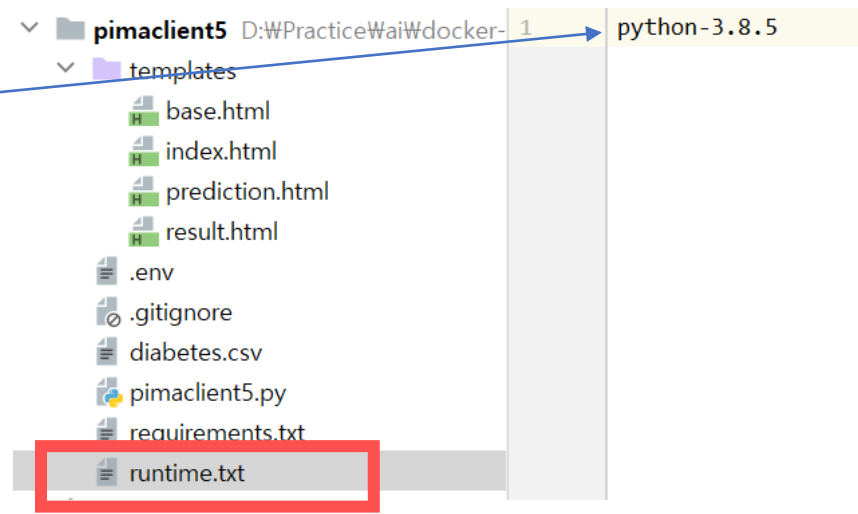
- **runtime.txt**: 프로그래밍 언어와 사용 버전
- **requirements.txt**: 파이썬 관련 라이브러리 의존성
- **Procfile**: 웹 애플리케이션을 구동하기 위해 실행되어야 하는 프로세스의 목록

requirements.txt, runtime.txt 파일 만들기:

(aisam) **pip freeze > requirements.txt** (다음 페이지 내용 참조)

(aisam) **python --version**

python-3.8.5



requirements.txt:

```
cachetools==4.0.0
Click==7.0
dominate==2.4.0
Flask==1.1.1
Flask-Bootstrap==3.3.7.1
Flask-WTF==0.14.2
google-api-python-client==1.7.9
google-auth==1.11.0
google-auth-httpplib2==0.0.3
httpplib2==0.17.0
itsdangerous==1.1.0
Jinja2==2.11.0
joblib==0.14.1
MarkupSafe==1.1.1
numpy==1.18.1
pandas==0.25.3
pyasn1==0.4.8
pyasn1-modules==0.2.8
python-dateutil==2.8.1
pytz==2019.3
rsa==4.0
scikit-learn==0.22.1
scipy==1.4.1
six==1.14.0
sklearn==0.0
uritemplate==3.0.1
visitor==0.1.3
waitress==1.4.2
Werkzeug==0.16.1
WTForms==2.2.1
matplotlib==3.0.2
requests==2.25.1
```

Running a production web server

로컬에서 production server 환경으로 실행:

```
(aisam) pip install waitress
```

```
(aisam) waitress-serve --port 8000 pimaclient5:app
```

=> 로컬 서버에서 실행: <http://samserver:8000>

Adding a Procfile

Top-level(app root) directory에서:

Package.json 파일은 Heroku에게 '애플리케이션이 Python 애플리케이션이다'라고 말해 주기는 하지만, 그것을 시작하는 방법은 말해주지 않는다.

이를 위해 **Procfile**을 사용한다:

Procfile은 애플리케이션에 의해 사용된 프로세스 타입과 애플리케이션을 시작시키는데 사용되는 커맨드들을 선언하는데 사용된다.

Procfile 파일 만들기: (Procfile 파일명은 대소문자 구분, 확장자 없음, 보이는 그대로 파일명을 붙일 것!)

파일명: **Procfile**

```
web: waitress-serve --port=$PORT pimaclient5:app
```

Top-level(app root) directory에서:

.env 파일 만들기

```
FLASK_APP=pimaclient5.py  
FLASK_CONFIG=heroku
```

.gitignore 파일 만들기

```
# Dependency directory
```

```
venv
```

```
# Environment variables (.env 파일에 있는 내용은 공개되면 안되므로 .gitignore에 포함)
```

```
.env
```

```
.idea
```

```
.vscode
```

Deploying with git push

Heroku CLI 터미널에서 Flask web app -> Heroku에 호스팅하기:

```
> heroku login  
  
> cd my-project/  
  
> git init  
  
> heroku git:remote -a pimaclient5  
  
> git add .  
  
> git commit -am "make it better"  
  
> git push heroku master  
  
> heroku open
```


Deploying an Upgrade

내용 일부 수정 후에는:

```
(aisam) git init
```

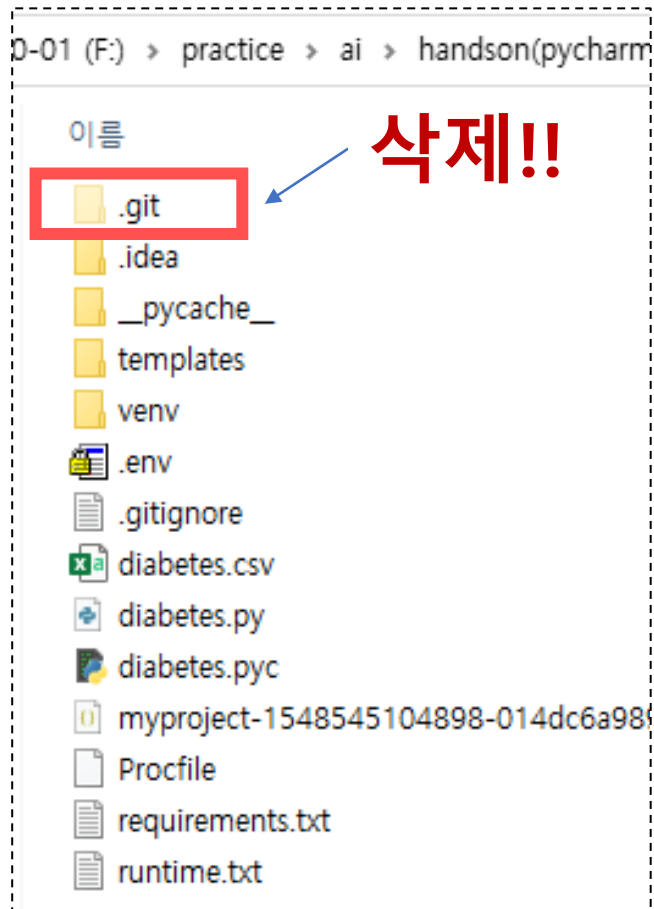
```
(aisam) git add .
```

```
(aisam) git commit -am "second..."
```

```
(aisam) git push heroku master
```

```
(aisam) heroku open
```

Git 사용 시 뭔가 잘못되었을 때 . . .



처음부터 다시 진행:

```
(aisam) git init
```

```
(aisam) git add .
```

```
(aisam) heroku git:remote -a pimaclient5
```

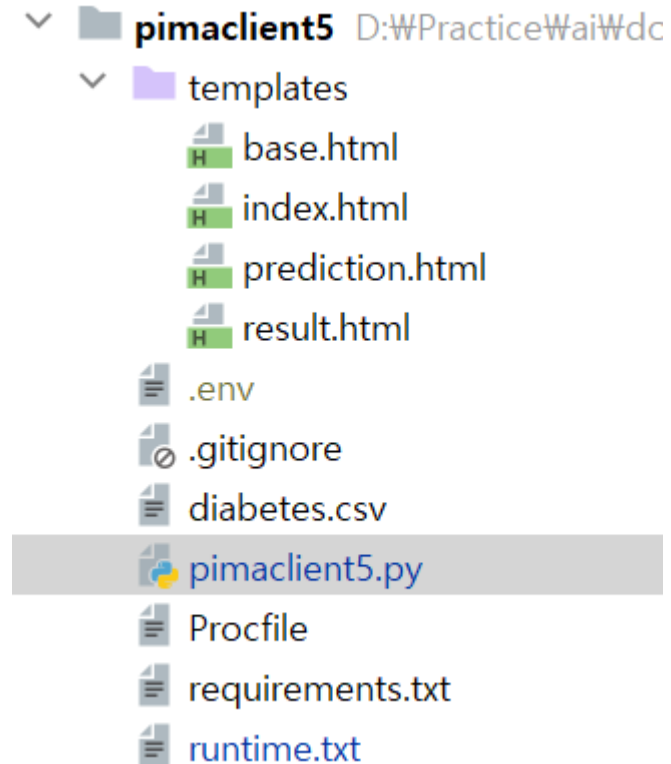
```
(aisam) git commit -am "first"
```

```
(aisam) git push heroku master
```

```
(aisam) heroku open
```

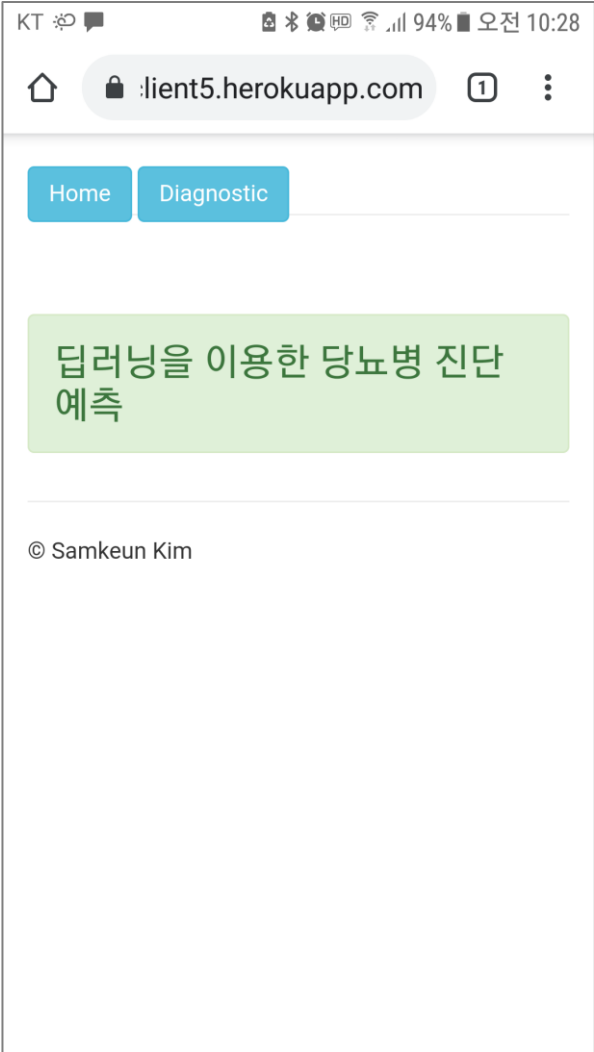
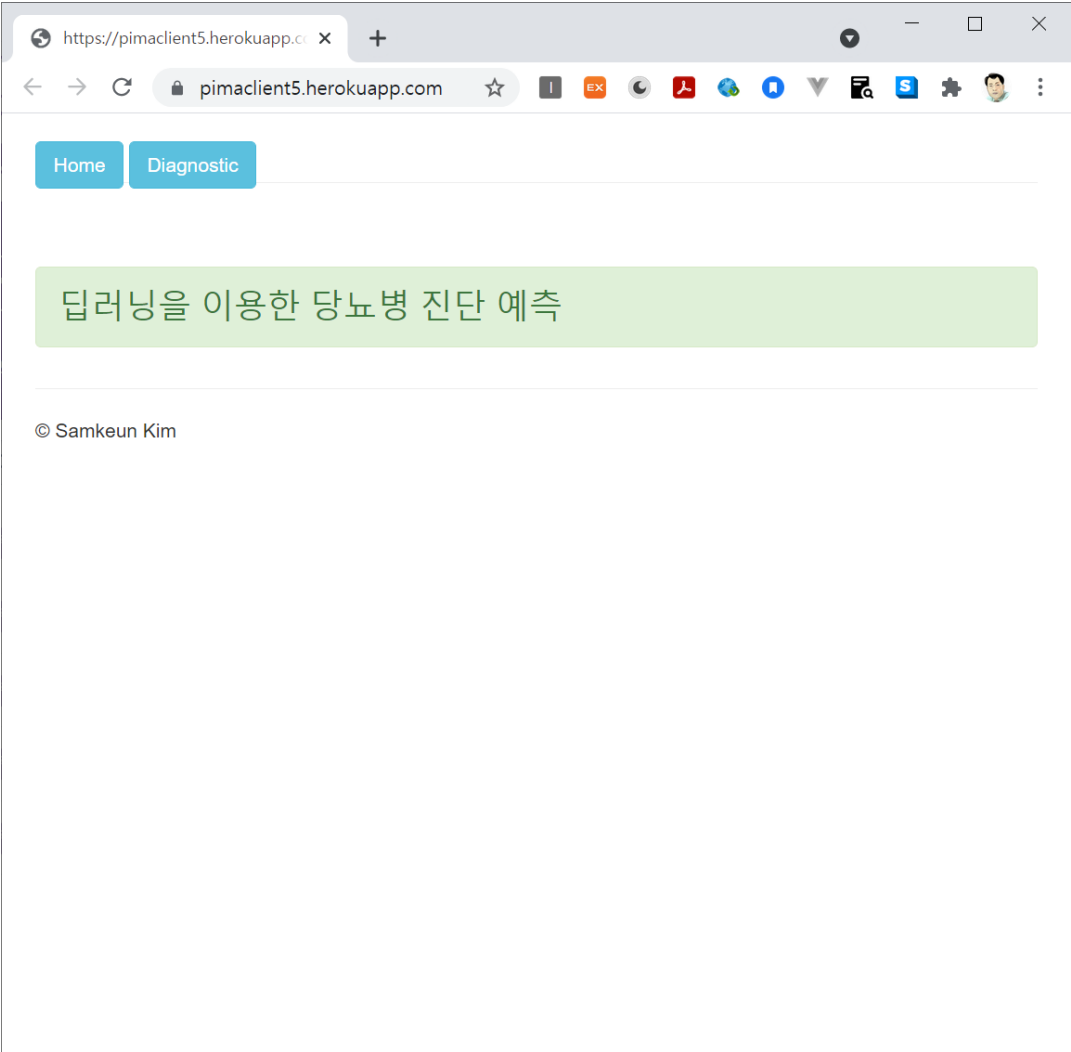
실습과제 17-1 딥러닝을 이용한 당뇨병 진단 예측 웹앱

Dynamic Flask Web App => Heroku Hosting 하기



소스 코드는 제 16강 강의자료 참조!!

실습과제 17-1 딥러닝을 이용한 당뇨병 진단 예측 웹앱



실습과제 17-1 딥러닝을 이용한 당뇨병 진단 예측 웹앱

https://pimaclient5.herokuapp.com/

pimaclient5.herokuapp.com

Home

Diagnostic

Fill in the medical data of the patient

Pregnancies

0

Glucose

137

Blood pressure

40

Skin thickness

35

Insulin

138

BMI

43.1

DPF Score

2.38

Age

33

Submit

© Samkeun Kim

KT

94%

오전 10:28

lient5.herokuapp.com

1

Home

Diagnostic

40

Skin thickness

35

Insulin

138

BMI

43.1

DPF Score

2.38

Age

33

Submit

© Samkeun Kim

실습과제 17-1 딥러닝을 이용한 당뇨병 진단 예측 웹앱

