

장고 차근차근 시작하기

Second Edition

당신의 파이썬/장고 페이스메이커가 되겠습니다. ;)

EP34. Heroku에 웹서비스 배포하기 (마지막 에피소드)

다양한 Platform as a Service

- Microsoft의 [Azure Web App for Containers](#)
- Google의 [AppEngine Flexible Environment](#)
- **[Heroku](#)**
- ZEIT의 [now.sh](#)

Heroku

AWS 플랫폼에서 서비스되고 있는 PaaS 플랫폼

- Plan

- Free

- 30분 동안 접속이 없으면, SLEEP
 - 550시간/월 제공 + (신용카드 정보 등록시) 450시간 추가 지원

- Hobby (\$7/월)

- Standard/Performance (\$25~)

- 주의

- Heroku 상의 디스크는 Persistent 스토리지가 아닙니다.
 - 매 프로세스 리부트마다 삭제됩니다.

가입하시고, Free + 550시간/월 얻기

The screenshot shows the Heroku account/billing page. The main content area displays 'Billing Information' with a message: 'This account does not have a credit card.' Below this is an 'Add Credit Card' button. A table shows 'Current Usage' as 'Not Available' and 'Platform Credits' as '\$0.00'. At the bottom, a section for 'Free Dyno Usage' shows '0.00 free dyno hours (0.00%) used this month' and '550.00 free dyno hours remaining this month'. A callout box on the right highlights the 'This account is billed to:' section, showing a 'VISA' card ending in 'XXXX'. It includes buttons for 'Change Credit Card' and 'Remove Credit Card'. Below this, it shows 'Current Usage' as 'Not Available' and 'Platform Credits' as '\$0.00'. At the bottom of the callout, it says 'Add your credit card to receive an additional 450 hours' and shows '0.00 free dyno hours (0.00%) used this month' and '1000.00 free dyno hours remaining this month'.

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Manage Account

Account Applications Billing

Billing Information

This account does not have a credit card.

Add Credit Card

Current Usage	Not Available
Platform Credits	\$0.00

Free Dyno Usage

Add your credit card to receive an additional 450 hours

0.00 free dyno hours (0.00%) used this month
550.00 free dyno hours remaining this month

Find out more

This account is billed to: VISA Visa ending in XXXX

Change Credit Card Remove Credit Card

Current Usage Not Available

Platform Credits \$0.00

Add your credit card to receive an additional 450 hours

0.00 free dyno hours (0.00%) used this month
1000.00 free dyno hours remaining this month

© 2018. All rights reserved by Ask Company

배포할 내용

- 장고 프로젝트는 gunicorn을 통한 구동
- static 파일은 [WhiteNoise](#)를 통한 직접 서빙
 - AWS S3/CDN를 통한 정적파일 관리를 추천
- 데이터베이스
 - Heroku 기본 지원 PostgreSQL 관리형 DB

배포 기본 설정 (공식문서)

[heroku-buildpack-python](#)을 통한 배포가 이뤄집니다.

파이썬 런타임 지정 ([공식문서](#))

- 지원하는 버전 (2018년 11월 현재)
 - python-3.7.0
 - python-3.6.6 (디폴트)
 - python-2.7.15
- 최상위 runtime.txt 파일에 지정
 - python-3.6.6

Database Provisioning ([Plan Tiers 공식문서](#))

- 자동 구성된 Heroku Postgres 데이터베이스에 대한 계정정보를 DATABASE_URL 환경변수를 통해 제공

Heroku Postgres tier	Downtime Tolerance	Fork	Follow	Rollback	HA	Disk Encryption
Hobby	< 4 hr downtime per mo.	No	No	No	No	No
Standard	< 1 hr downtime per mo.	Yes	Yes	4 days	No	Yes
Premium	< 15 min downtime per mo.	Yes	Yes	1 week	Yes	Yes
Private	< 15 min downtime per mo.	Yes	Yes	1 week	Yes	Yes
Shield	< 15 min downtime per mo.	Yes	Yes	1 week	Yes	Yes

Procfile 파일 생성

gunicorn을 통한 기본옵션 실행

```
web: gunicorn askcompany.wsgi --log-file -
```

설치할 라이브러리 명시

- 최상위 requirements.txt 에 명시하면, Heroku에서 자동 수행
- Heroku 필수 패키지
 - [django-heroku](#)
 - **장고 2.0 이상**을 지원
 - DATABASE_URL 환경변수 처리, 로깅 설정, staticfiles 처리, ALLOWED_HOSTS에 ["*"] 추가, whitenoise를 통한 정적파일 처리, Heroku CI 설정
 - whitenoise
 - 정적파일 직접 서빙
 - gunicorn 혹은 uwsgi
 - Python WSGI HTTP Server
 - psycopg2
 - PostgreSQL Driver

django-database-url 활용 예시

```
>>> import django_database_url
```

```
>>> django_database_url.parse('postgres://u-----:w-----@ec2-111-222-333-444.'  
                             'compute-1.amazonaws.com:5431/d8r82722r2kuvn')
```

```
{'CONN_MAX_AGE': 0,  
 'ENGINE': 'django.db.backends.postgresql_psycopg2',  
 'HOST': 'ec2-111-222-333-444.compute-1.amazonaws.com',  
 'NAME': 'd8r82722r2kuvn',  
 'PASSWORD': 'w-----',  
 'PORT': 5431,  
 'USER': 'u-----'}
```

django-heroku를 통한 처리 ([관련코드](#))

```
from .common import *  
  
import django_heroku  
django_heroku.settings(locals())
```


WhiteNoise를 통한 static 서빙 ([공식문서](#))

- django-heroku를 통한 자동 셋업
- 이제, Heroku 배포 중에 자동으로 다음 명령이 수행
 - 쉘> `python manage.py collectstatic --noinput`
- 디버깅 옵션 켜기 : 쉘> `heroku config:set DEBUG_COLLECTSTATIC=1`
- collectstatic 끄기 : 쉘> `heroku config:set DISABLE_COLLECTSTATIC=1`

Heroku CLI를 통한 배포

Heroku CLI 설치


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

 **macOS**

Download the installer

Also available via Homebrew:


```
$ brew install heroku/brew/heroku
```

 **Windows**

Download the appropriate installer for your Windows installation:

64-bit installer

32-bit installer

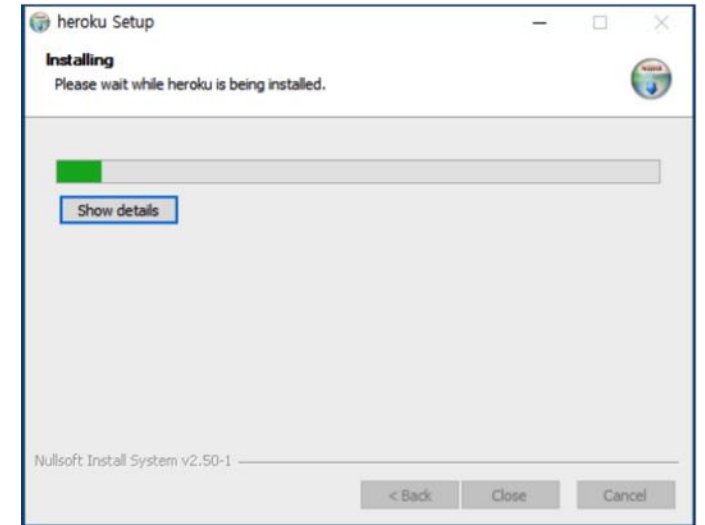
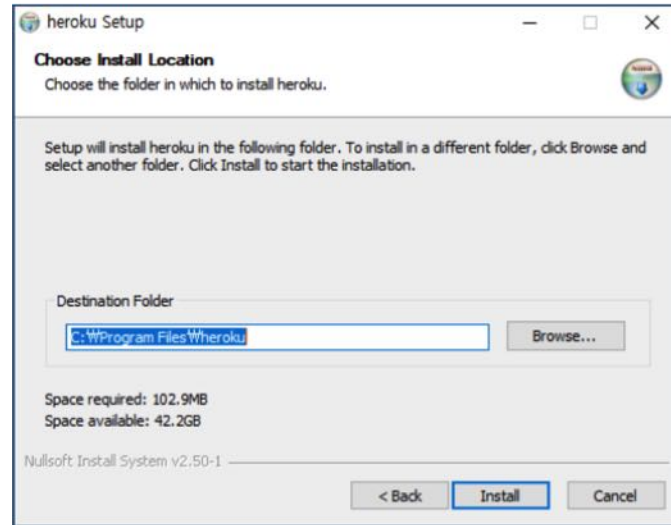
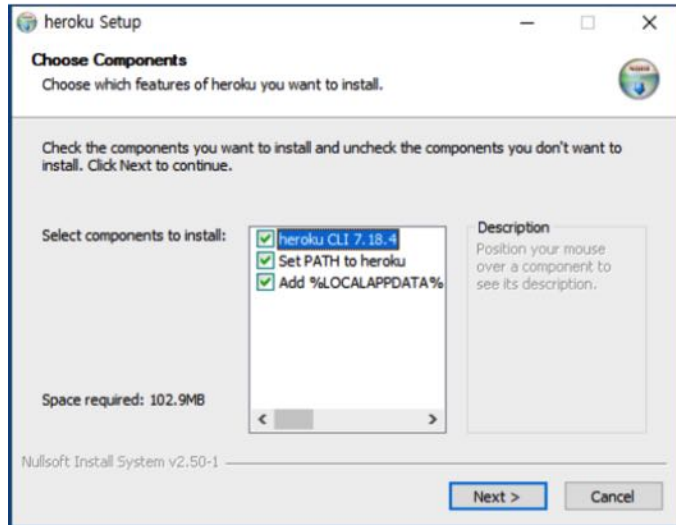
 **Ubuntu 16+**

Run the following from your terminal:

```
$ sudo snap install --classic heroku
```

Snap is available on other Linux OS's as well.

Heroku 설치 화면 (샘플)



```
C:\Users\allieus> heroku --version
heroku/7.18.4 win32-x64 node-v11.0.0
```

```
→ heroku --version
heroku/7.18.4 darwin-x64 node-v11.0.0
```


현재 프로젝트를 git 저장소로 변경

- 프로젝트가 정리가 덜 되었다면,
 - <https://github.com/allieus/askdjango-s2-basic> 프로젝트를 clone 떠주세요.
- Git GUI 프로그램
 - SourceTree : <https://www.sourcetreeapp.com/>
 - GitHub Desktop : <https://desktop.github.com/>
 - GitKraken : <https://www.gitkraken.com/>

Heroku CLI를 통한 인증 수행

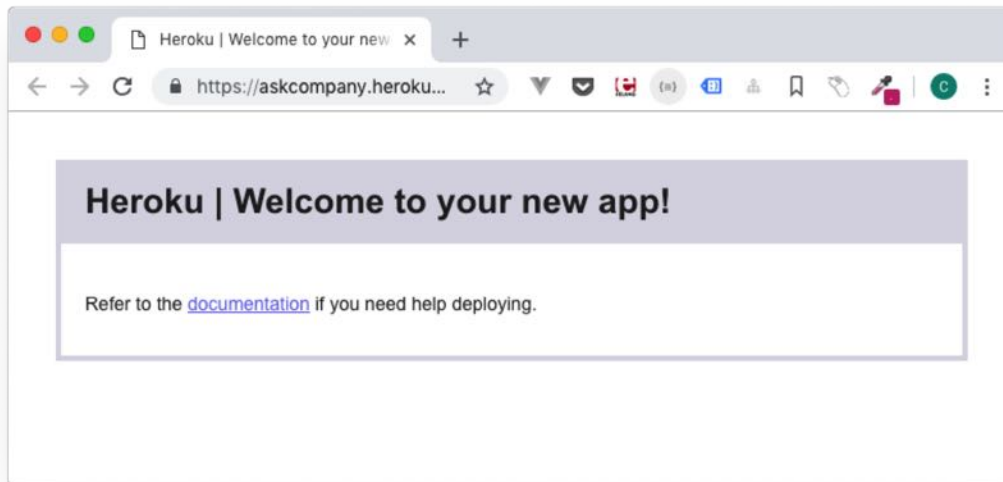
- 셸 > heroku login

```
C:\Users\allieus> heroku login
heroku: Enter your login credentials
Email: [REDACTED]
Password: *****
Logged in as [REDACTED]
C:\Users\allieus>
```

Heroku CLI를 통해 프로젝트 생성

```
→ heroku apps:create askcompany  
Creating askcompany... done  
https://askcompany.herokuapp.com/ | https://git.heroku.com/askcompany.git
```

- Heroku 기본 프로젝트 생성
 - 생성된 *.herokuapp.com 주소로 접속해보세요.



Git 저장소 remote로 heroku 추가됨

```
14 [remote "heroku"]  
15   url = https://git.heroku.com/askcompany.git  
16   fetch = +refs/heads/*:refs/remotes/heroku/*
```

필요한 환경변수 설정

- 셸> **heroku config:set** DJANGO_SETTINGS_MODULE=askcompany.settings.?????????

배포 수행 (1)

- 셸 > git push heroku master

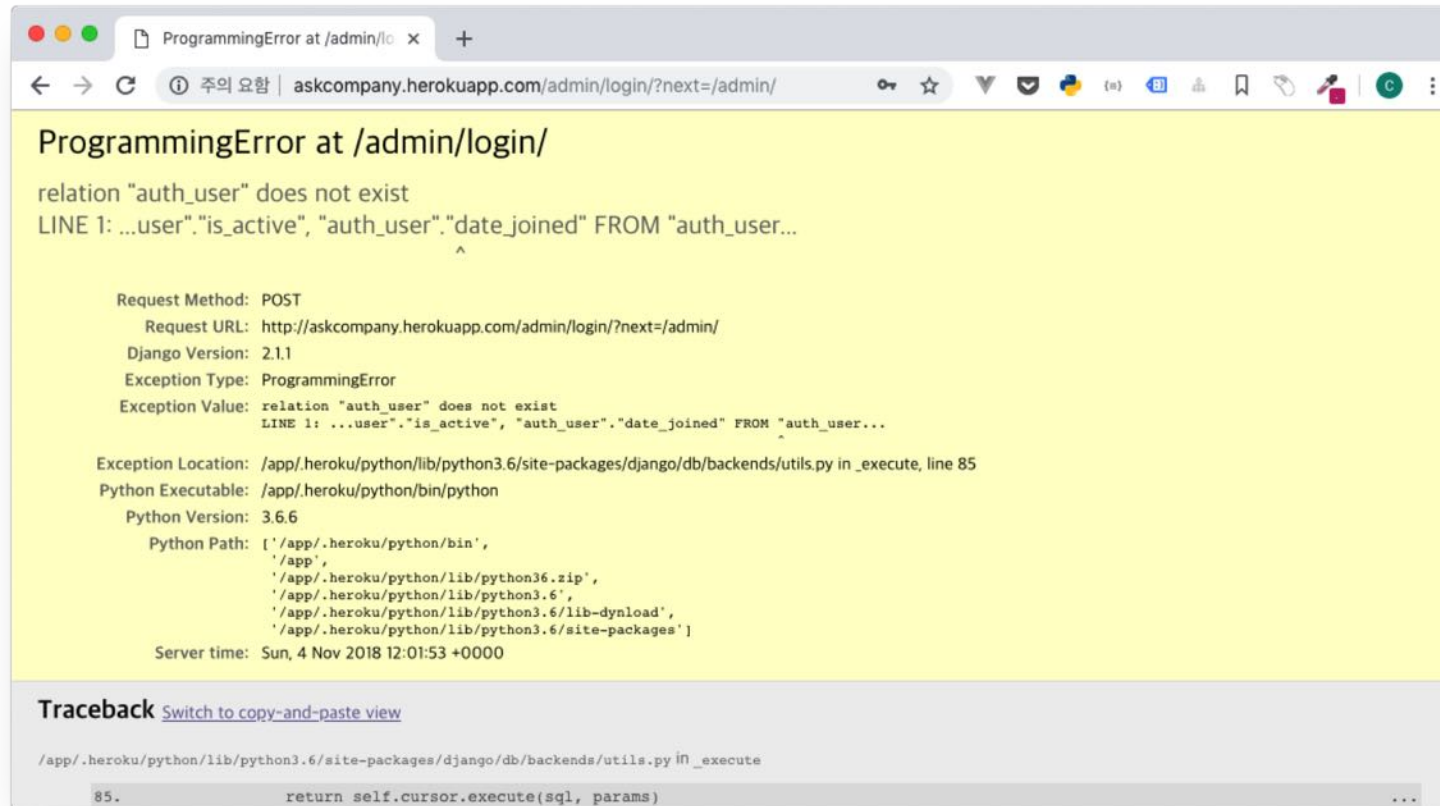
```
→ git push heroku master
Counting objects: 181, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (155/155), done.
Writing objects: 100% (181/181), 19.32 KiB | 1.07 MiB/s, done.
Total 181 (delta 86), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
remote:          Using supported version of Python 3.6 (python-3.6.6)
remote: -----> Installing python-3.6.6
remote: -----> Installing pip
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote:          Collecting django==2.1.1 (from -r /tmp/build_5a5d24a3b
```

배포 수행 (2)

```
remote:          Running setup.py install for MarkupSafe: finished with status 'done'
remote:          Successfully installed MarkupSafe-1.0 django-2.1.1 django-debug-toolbar-1.10.1 d
a-2.4.1 jinja2-2.10 pillow-5.3.0 pytz-2018.7 six-1.11.0 sqlparse-0.2.4
remote:
remote: -----> $ python manage.py collectstatic --noinput
remote:          145 static files copied to '/tmp/build_5a5d24a3bd11e25407fcbec16a5cb680/static'.
remote:
remote: -----> Discovering process types
remote:          Procfile declares types -> (none)
remote:
remote: -----> Compressing...

remote:          Done: 53.4M
remote: -----> Launching...
remote:          Released v5
remote:          https://askcompany.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/askcompany.git
 * [new branch]      master -> master
```

웹페이지에 접속해보면???



ProgrammingError at /admin/login/

relation "auth_user" does not exist
LINE 1: ...user"."is_active", "auth_user"."date_joined" FROM "auth_user..."

Request Method: POST
Request URL: http://askcompany.herokuapp.com/admin/login/?next=/admin/
Django Version: 2.1.1
Exception Type: ProgrammingError
Exception Value: relation "auth_user" does not exist
LINE 1: ...user"."is_active", "auth_user"."date_joined" FROM "auth_user..."
Exception Location: /app/.heroku/python/lib/python3.6/site-packages/django/db/backends/utils.py in _execute, line 85
Python Executable: /app/.heroku/python/bin/python
Python Version: 3.6.6
Python Path: ['/app/.heroku/python/bin',
'/app',
'/app/.heroku/python/lib/python3.6.zip',
'/app/.heroku/python/lib/python3.6',
'/app/.heroku/python/lib/python3.6/lib-dynload',
'/app/.heroku/python/lib/python3.6/site-packages']
Server time: Sun, 4 Nov 2018 12:01:53 +0000

Traceback [Switch to copy-and-paste view](#)

```
/app/.heroku/python/lib/python3.6/site-packages/django/db/backends/utils.py in _execute
85.         return self.cursor.execute(sql, params)
```

migrate 수행 (1)

- heroku 쉘 구동

```
→ heroku run bash
Running bash on askcompany... up, run.9707 (Free)
~ $ python --version
Python 3.6.6
```

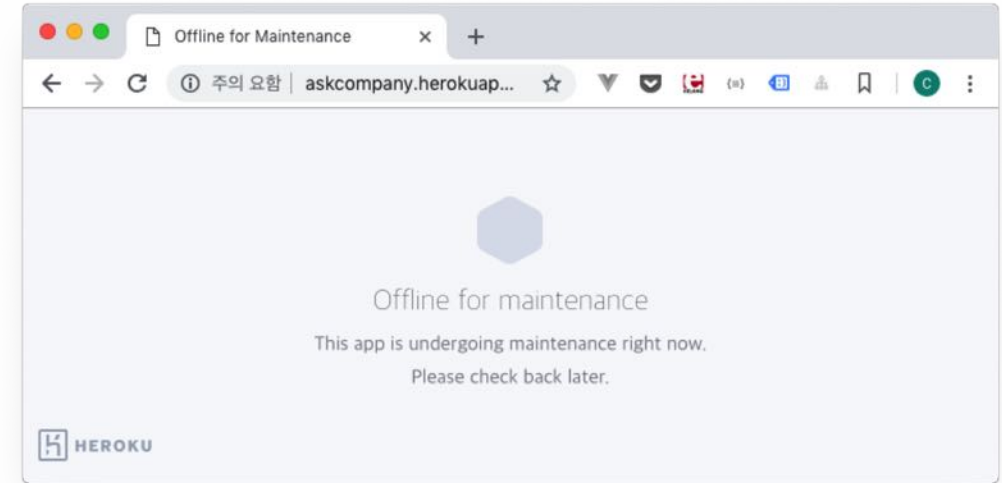
```
~ $ python manage.py --version
2.1.1
~ $ python manage.py showmigrations
admin
[ ] 0001_initial
[ ] 0002_logentry_remove_auto_add
[ ] 0003_logentry_add_action_flag_choices
auth
[ ] 0001_initial
[ ] 0002_alter_permission_name_max_length
[ ] 0003_alter_user_email_max_length
[ ] 0004_alter_user_username_opts
[ ] 0005_alter_user_last_login_null
[ ] 0006_require_contenttypes_0002
[ ] 0007_alter_validators_add_error_messages
[ ] 0008_alter_user_username_max_length
[ ] 0009_alter_user_last_name_max_length
blog
[ ] 0001_initial
[ ] 0002_comment
[ ] 0003_post_author_name
[ ] 0004_post_photo
common
(no migrations)
contenttypes
[ ] 0001_initial
[ ] 0002_remove_content_type_name
sessions
[ ] 0001_initial
shop
[ ] 0001_initial
[ ] 0002_item_is_publish
[ ] 0003_item_photo
```


migrate 수행 (2)

```
~ $ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, contenttypes, sessions, shop
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying blog.0001_initial... OK
  Applying blog.0002_comment... OK
  Applying blog.0003_post_author_name... OK
  Applying blog.0004_post_photo... OK
  Applying sessions.0001_initial... OK
  Applying shop.0001_initial... OK
  Applying shop.0002_item_is_publish... OK
  Applying shop.0003_item_photo... OK
~ $ python manage.py createsuperuser
Username (leave blank to use 'u4061'): allieus
Email address:
Password:
Password (again):
Superuser created successfully.
~ $
~ $ exit
```

그 외 명령

- Maintenance 켜기
 - 쉘> heroku maintenance:on
- Heroku App 삭제
 - 쉘> heroku apps:delete
- 로그 확인
 - 쉘> heroku logs -tail
- 도메인 설정
 - 쉘> heroku domains
- 그 외
 - 쉘> heroku --help



```
→ heroku apps:delete
  ▶ WARNING: This will delete askcompany including all add-ons.
  ▶ To proceed, type askcompany or re-run this command with --confirm askcompany

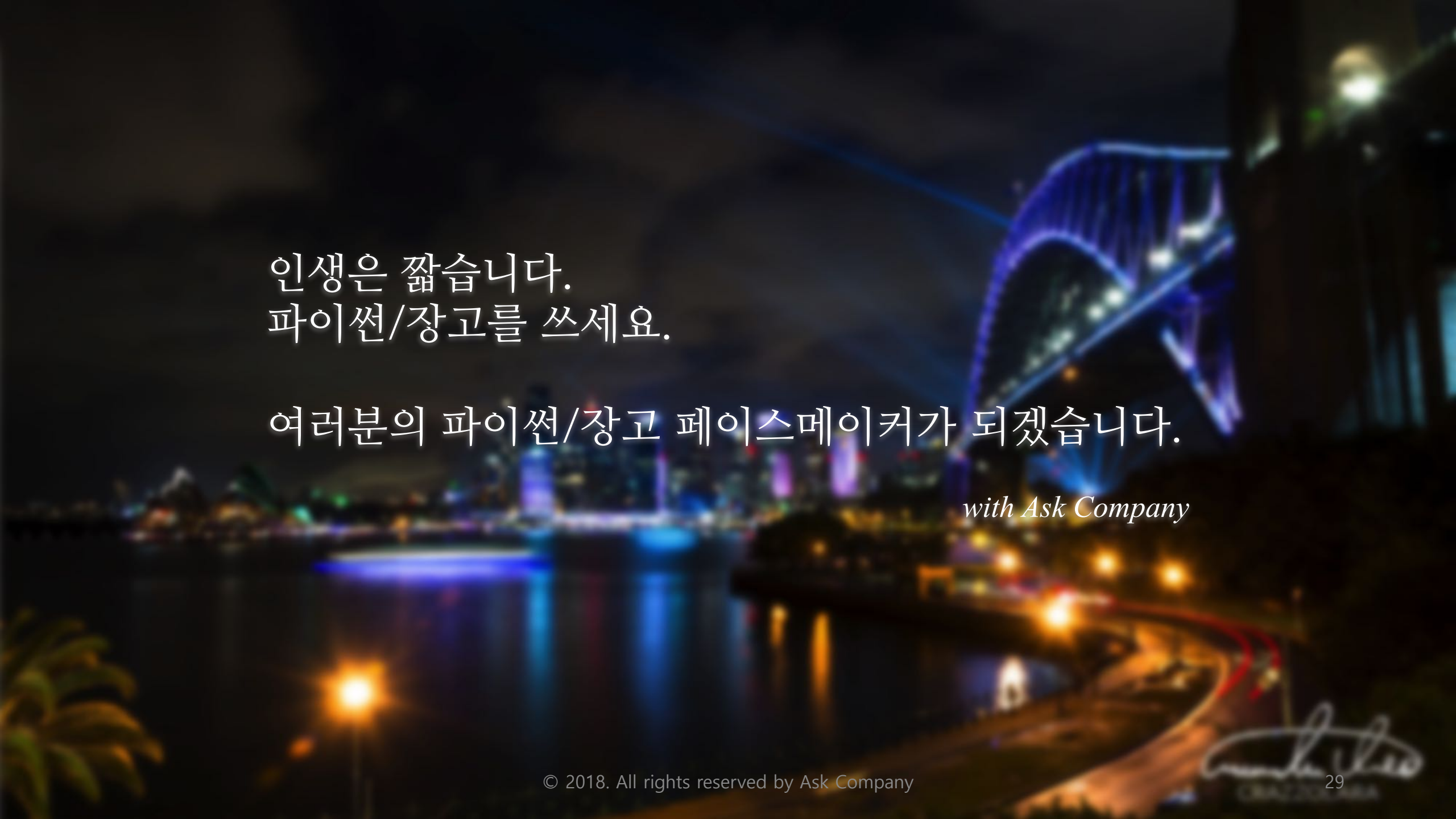
> askcompany
Destroying askcompany (including all add-ons)... done
```

보충하면 좋을 내용.

- Media 파일 처리
 - AWS S3와 연동해보세요.
- 30분 sleep에 빠지지않게 할려면,
 - 적어도 29분 주기로 웹페이지 접속이 있어야 합니다.
 - [Azure Scheduler](#)를 통해 Free로도 HTTP 요청을 보낼 수 있습니다.

Heroku에는 많은 Feature들이 있습니다.

- 공식문서
 - <https://devcenter.heroku.com/categories/reference>

A nighttime photograph of a cityscape featuring a bridge with blue and white lights. The lights are reflected in the water below. The sky is dark, and the overall scene is illuminated by the city lights.

인생은 짧습니다.
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

with Ask Company