

프로덕션 배포하기: wsgi, cloud computing service, cron jobs

Instructors:

김찬욱 @narayo9

Assignment 3 리뷰

목차

How to deploy django?

Cloud computing service?

cron job의 필요성과 custom management
command

Assignment 4 소개

How to deploy django?

Production django는 무엇이 다른가
메인 출처: [django 공식 문서](#)

웹 서버, 인터페이스?

python manage.py runserver는 production에서 사용 불가
(퍼포먼스, 디버깅)



웹 프레임워크인 Django는 작동하기 위해 웹 서버(apache, nginx 등)가 필요함.

그리고 대부분의 웹 서버는 기본적으로 Python을 사용하지 않음

그러한 커뮤니케이션이 이루어지도록 인터페이스가 필요 => wsgi, asgi의 필요성

인터페이스: wsgi vs asgi

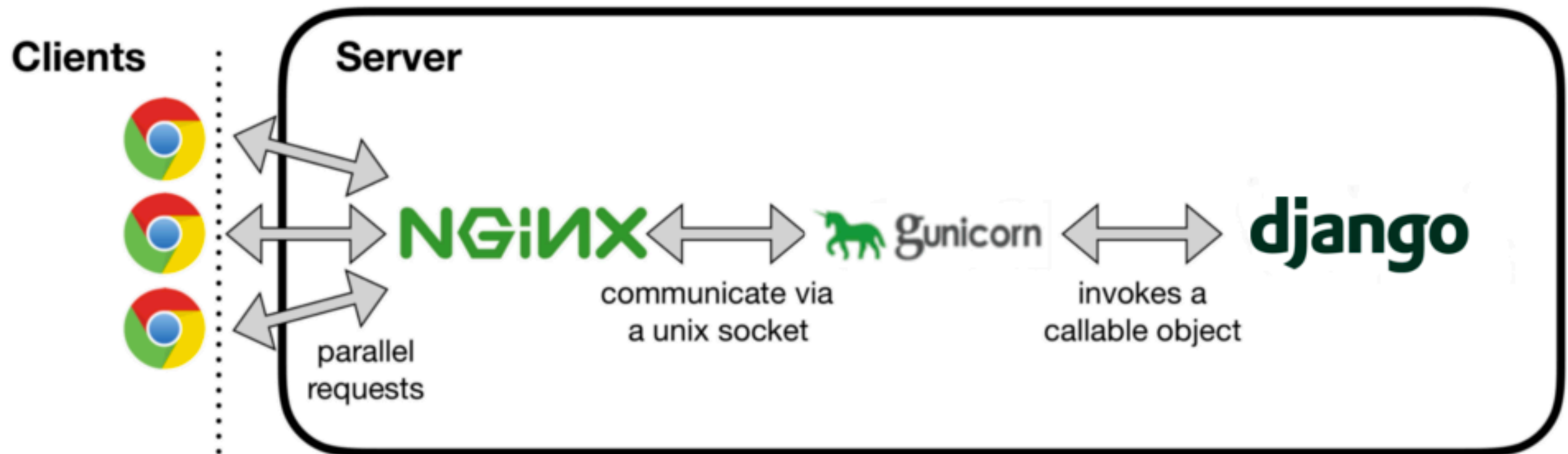
Django는 현재 WSGI와 ASGI의 두 가지 인터페이스를 지원

WSGI: 동기 파이썬 웹 서버 인터페이스 표준 => 오늘 사용

ASGI: 비동기 친화적인 표준

웹 개발의 미래: 비동기가 궁금하다면?

표준 배포 구조 소개



웹서버(Nginx)가 꼭 필요해?

출처

Although there are many HTTP proxies available, we strongly advise that you use Nginx. ...

Without this buffering Gunicorn will be easily susceptible to denial-of-service attacks.

실습: <myproject>/wsgi.py 파일 확인

두말할 필요 없이 직접 해보자.

링크

조언: 모르겠는 게 많겠지만, 최대한 기본값을 따라가고 한 레이어 한 레이어 왜 필요한지 생각해 보세요

실습: gunicorn으로 실행해보기

링크

뭐가 다른걸까?

Deployment checklist

링크

```
python manage.py check --deploy
```

실제로는 하나 하나 읽어보고 해결할 것

중요한 세팅 몇 가지를 살펴보자!

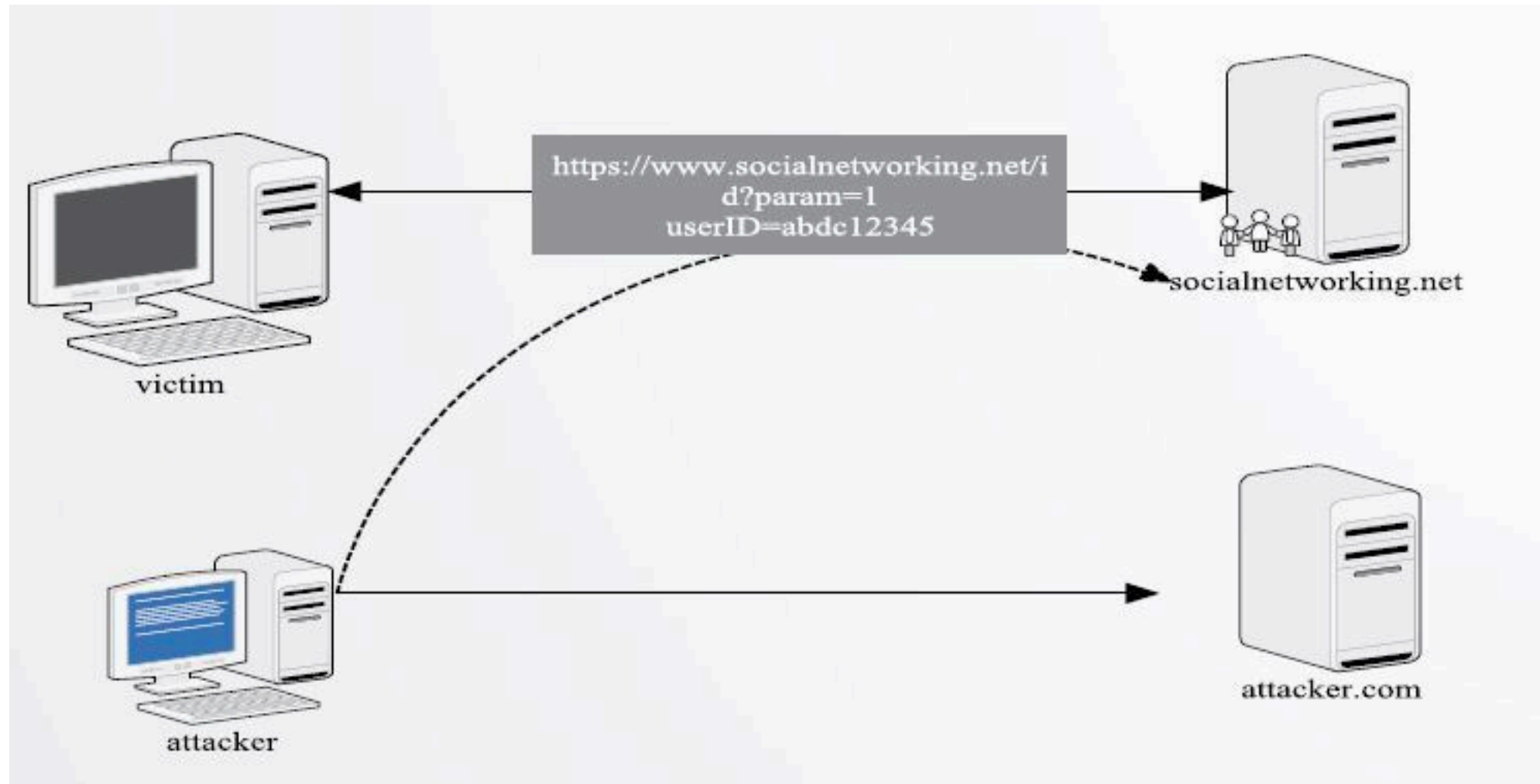
Production settings

DEBUG: Never deploy a site into production with DEBUG turned on. if DEBUG is False, you also need to properly set the ALLOWED_HOSTS setting.

ALLOWED_HOSTS: A list of strings representing the host/domain names that this Django site can serve.

CSRF 공격 방지

읽을거리: CSRF(Cross Site Request Forgery)?



출처: **CSRF 공격이란? 그리고 방어 기법**

Cloud computing service

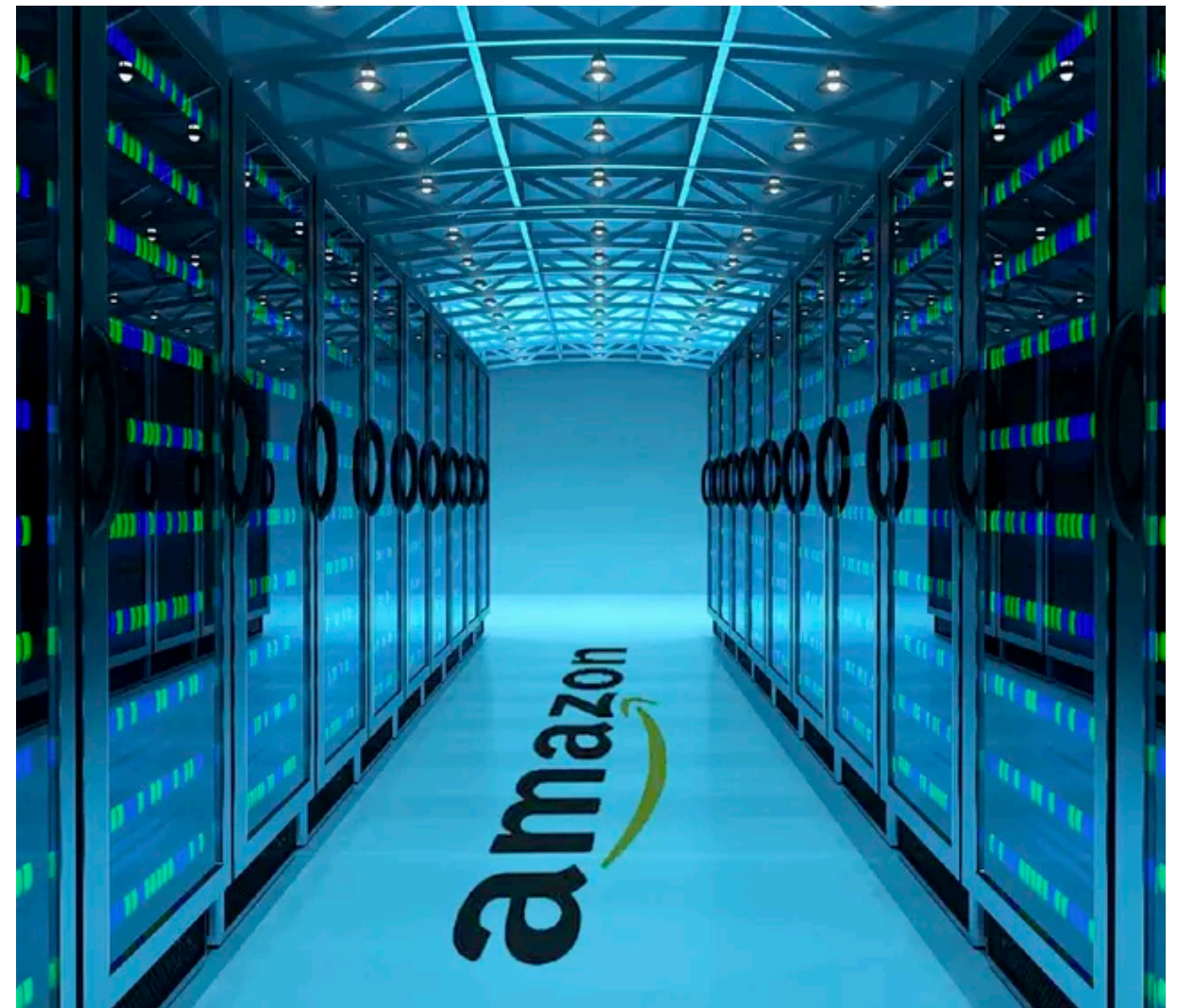
부제: 참 편한 세상에 살고 있다

Cloud computing service 의 필요성

클라우드 컴퓨팅 플랫폼

힘들게 IDC를 직접 설치할 수
고 없이, 서버를 사용할 수 있음

클라우드 컴퓨팅 서비스(EC2,
Elastic beanstalk)
스토리지 서비스(S3, ...)
데이터베이스 서비스
(RDS, ...)

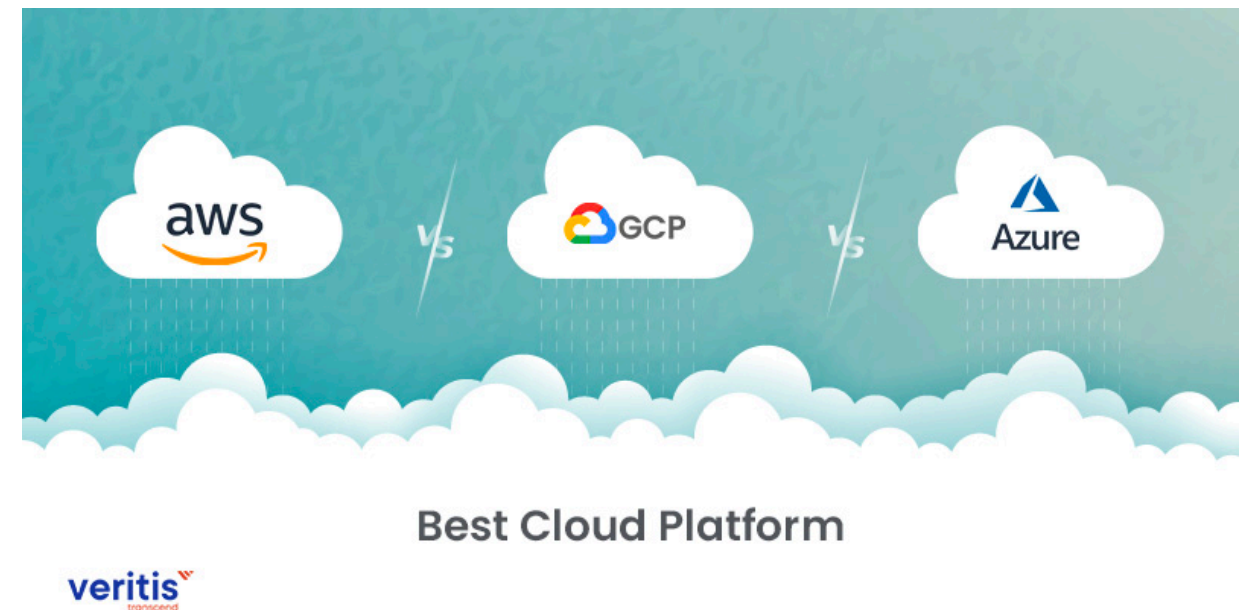


읽을거리: AWS vs GCP vs Azure

실제로 회사에 들어가면
쓰게 될 Cloud
computing services

링크

Devops / infra



읽을거리: iaas / paas / saas

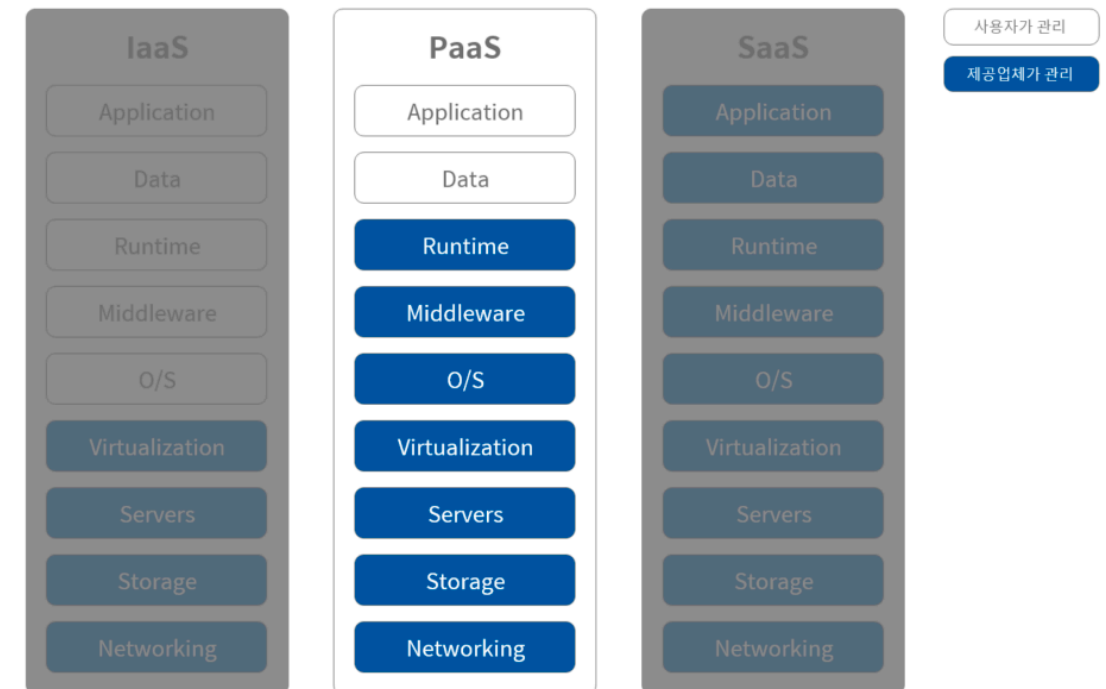
링크

iaas인 AWS를 직접 다뤄보고 싶었지만...



paas인 render.com을 사용해봅시다!

기존에 큰 강자로는 heroku가 있었음



실습: render.com에 django project 올려보기

가이드 문서

Django 3.2~ 사용하기

Django logging 켜는 설정 =>

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'handlers': {
        'console': {
            'level': 'DEBUG',
            'class': 'logging.StreamHandler',
        },
    },
    'root': {
        'level': 'INFO',
        'handlers': ['console']
    },
}
```

cron job의 필요성과 custom management command

[Django 공식 문서](#)

Cron job의 등장

현재 작성한 모든 django code는 request triggered

이런 작업들은 어떻게 할 수 있을까?

매일 새벽, 오늘의 인기 순위를 계산해 저장해주세요

마지막 로그인 한 지 90일 된 유저를 비활성화해주세요

Cron job: schedule triggered

매일 새벽, 오늘의 인기 순위를 계산해 저장해주세요

마지막 로그인 한 지 90일 된 유저를 비활성화해주세요

crontab.guru

구현체: crontab, render.com cronjobs

crontab: unix 기반 app에서 service로 실행됨,
파일에 job 지정해두는 구조

render.com에서 cronjobs 기능 제공

Management command?

여러분이 `manage.py` 뒤에 붙였던 그것들
Cron job으로 유용하게 활용할 수 있다!

```
python manage.py runserver  
python manage.py migrate  
python manage.py sqlmigrate  
...
```

읽을거리: 우리가 사용했던 **command** 살펴보기

command+shift+o

migrate.py

runserver.py

makemigrations.py

management command 만드는 법

To do this, add a management/commands directory to the application. Django will register a manage.py command for each Python module in that directory whose name doesn't begin with an underscore. For example:

```
polls/  
  __init__.py  
  models.py  
  management/  
    __init__.py  
    commands/  
      __init__.py  
      _private.py  
      closepoll.py  
  tests.py  
  views.py
```

실습: 90일이 지난 유저 비활성화 command 추가

예시

Management command arguments

[Python argparse package](#) 참고!

공식 문서

실습: render.com에 cronjob 추가

Assignment 4 소개

거의 다 왔다...

Questions?