

19.03.20 리눅스에 웹 올리기 (메뉴얼-설치 부터 nginx+gunicorn)

노트북: [TIL-MY]

만든 날짜: 2019-03-22 오후 4:55

작성자: 황인범

URL: <https://github.com/pypa/pip/issues/5221>

그전에 가상머신 만들 때 ubuntu로 이름을 짓고 메모리 4에 bridge방식으로 net을 연결한다.

1. sudo apt-get install git / git clone

https://hwanginbeom@bitbucket.org/hwanginbeom/smtr_aws_web.git

2. sudo apt-get install mysql-server로 mysql을 설치하고 smtr_dev 데이터베이스를 생성한다.

```
sudo mysql -u root
```

```
1. DROP USER 'root'@'localhost';
2. CREATE USER 'root'@'%' IDENTIFIED BY '';
3. GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
4. FLUSH PRIVILEGES;
5. create database smtr_dev
6. git clone을 통해 받은 디렉토리에 들어가 sudo apt-get install python python-
   pip python3-venv 세가지를 받는다.
```

3. 기존에 있던 smtr_venv를 삭제한다.

4. python3 -m venv smtr_venv 한 후 venv에 들어가 가상환경에서 pip install -r requset.txt 를 진행한다.

5. python manage.py makemigrations

6. python manage.py migrate

7. mysql에 들어가 question과 scope 부분 한글 데이터 받을 수 있게 끔

```
# alter table smtrapp_examscope /
smtrapp_question convert to charset utf8 / 로 두개 바꿔준다
# sql data 넣어준다.
```

8. python manage.py runserver 0.0.0.0:8000

9. 127.0.0.1:8000 으로 접속

10. sudo apt-get install gunicorn

11.

```
gunicorn project.wsgi:application --bind=127.0.0.1:8001 --
daemon --reload
```

12. 구동 테스트:

13. `unicorn --bind 0.0.0.0:8000 smtrsite.wsgi:application`
 14. 가상환경나오기
 15. 서비스등록용 스크립트 작성
 16. `/etc/systemd/system/gunicorn.service` 파일을 아래와 같은 내용으로 생성
 17. 만약 구동 실패시 에러 로그는 `/var/log/syslog` 에서 확인
 18. `ps -ef | grep gunicorn`
-

1. `sudo apt install nginx`로 nginx 설치
2. `sudo vi /etc/nginx/sites-available/smtrsite` -> 사용가능한 사이트 설정을추가한다.
- 3.

```
server {
    listen 80; <-nginx만의 포트
    server_name 192.168.0.131; --> 사이트에 보이는 내용. 보통은 IP, 도메인
    기록
    location = /favicon.ico { access_log off; log_not_found off; } -->
    파비콘없어도 에러 안나게
    location /static/ {
        root /home/ubuntu/smtr_aws_web;    --> 프로젝트폴더,
    }
    location / {
        include proxy_params;
        proxy_pass
        http://unix:/home/ubuntu/smtr_aws_web/run/gunicorn.sock;    --> gunicorn 소
        켓화일 있는곳
    }
}
```

4. `sudo ln -s /etc/nginx/sites-available/django_test /etc/nginx/sites-enable` 사이트 추가
 5. `sudo nginx -t` 설정 문법 검사
 6. `sudo systemctl restart nginx` 재기동
 7. `sudo ufw delete allow 80` 방화벽 해제
 8. `sudo ufw allow 'Nginx Full'`
 9. `nginx -s quit`
 10. `sudo systemctl restart nginx` 재기동
 11. `ps ax | grep gunicorn`하고 `kill gunicorn`
-

3. 에러 부분 pip 를 사용 할 때

from pip import main 이라고 뜨면서 안될 때

venv에서 나간후 pip를 삭제하고 reinstall 을 한다.

First, some notes about what happened here on Debian/Ubuntu (and likely a few more Linux Distros):

- pip does not support the use of it's internals by importing it. More on that in the documentation [here](#).

- Debian (hence Ubuntu) does not support modifying their package manager managed files using something, that well, isn't their package manager.

This issue is caused by, well, both being violated in a way.

- Debian uses the pip's internal methods (which no longer work due to a reorganization of pip's internals). Debian's assuming here that the pip version in their repositories is the one that would be installed.
- Running `pip install --upgrade pip`, as root, without any other parameters modifies files that are supposed to be managed by apt, which breaks the script by Debian.

Some general tips on Linux:

- It's a good habit to use `--user` whenever outside a venv.

```
pip install --upgrade --user pip
```

- *Never* run pip with sudo unless you know what you're doing.

What's the workaround?

[@standag's solution](#) is useful when this is caused by bash's caching of executables.

```
hash -r pip # or hash -d pip
```

If you've modified your OS package manager's installation of pip (eg by using `sudo pip`) and `python -m pip` is still working, one workaround is to uninstall the pip installed version and reinstall the package manager installed version.

```
python -m pip uninstall pip # this might need sudo
sudo apt install --reinstall python-pip
```

If you're not on Debian/Ubuntu *and* pip broke for you, try running:

```
python -m pip install --force-reinstall pip
```

If the above this doesn't resolve your problems, please file a new issue.

#5. manage를 사용하는데 있어서 from exec 가 계속 뜬다면 venv 를 만들어 실행해야된다.

```
python3 -m pip install virtualenv
virtualenv venv -p python3
하고 venv에서 pip install -r request.txt를 한다.
```

#6 번 과정에서 퍼미션 관련 에러가 난다면 setting에 들어가서 비밀번호를 치는 부분을 " 이렇게 공백으로 만든다. 그전에 선례될 게 이 부분을 한다.

1번 받고

2번 권한 새로 설정하고

```
sudo mysql -u root
DROP USER 'root'@'localhost';
CREATE USER 'root'@'%' IDENTIFIED BY '';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
FLUSH PRIVILEGES;
```

3번 `python manage.py makemigrations smtrapp`

4번 `python manage.py migrate smtrapp`
