

Django 배포 방법

django project 이름을 *mysite*로 가정

```
scp -i key -r mysite ubuntu@ip:/home/ubuntu/mysite
```

```
# anaconda 가상환경과 python 가상환경 중 하나 선택해서 사용 (anaconda 사용이 더 편함)
# 가상환경 이름을 mysite로 가정

# anaconda venv
conda create -n mysite python=3.7 django
conda activate mysite

# python venv
sudo apt install python3-venv -y
mkdir venv
cd venv
python3 -m venv mysite
cd mysite/bin
# .과 activate 사이 공백
. activate

cd ~
```

django 설정

```
cd mysite
python manage.py migrate

# config directory로 이동
vim settings.py

# aws ip 넣어주기
ALLOWED_HOSTS = ['13.114.200.183']
```

django 만으로 실행

```

cd ~/mysite

# nohup : terminal 연결이 끊어져도 계속 실행
# & : background 실행
nohup python manage.py runserver 0:8000 &

# 종료방법 : ps 명령으로 python pid 확인 후 kill (pid는 실행 때 마다 바뀜)
ps
kill 8564

# 서버 로그 확인 방법
cat nohup.out

```

web server와 분리 후 실행

```

# config directory로 이동
vim settings.py
# True -> False
DEBUG=False

```

gunicorn 설정

```
pip install gunicorn
```

```
vim mysite.env
```

```
DJANGO_SETTINGS_MODULE=config.settings.prod
```

```
vim /etc/systemd/system/mysite.service
```

```

[Unit]
Description=gunicorn daemon
After=network.target

[Service]
User=ubuntu
Group=ubuntu
WorkingDirectory=/home/ubuntu/mysite
ExecStart=/home/ubuntu/anaconda3/envs/mysite/bin/gunicorn \
--workers 2 \
--bind unix:/tmp/gunicorn.sock \
config.wsgi:application

[Install]
WantedBy=multi-user.target

```

```
# 서비스 시작
sudo systemctl start mysite.service
# 서비스 확인
sudo systemctl status mysite.service
```

nginx 설정

```
sudo apt install nginx -y
cd /etc/nginx/sites-available
```

```
sudo vim mysite
```

```
server {
    listen 80;
    server_name 13.114.200.183;

    location /static {
        alias /home/ubuntu/mysite/static;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/tmp/gunicorn.sock;
    }
}
```

```
cd /etc/nginx/sites-enabled
sudo rm default
sudo ln -s /etc/nginx/sites-available/mysite
```

```
# nginx 재시작
sudo systemctl restart nginx
```

```
# 13.114.200.183 접속해보기!
```