

```

1 Lab. nginx 연결하기
2   ※ Ubuntu Linux 기준
3
4 1. nginx와 uwsgi 연결하기
5   1)nginx를 설치한다.
6     $ sudo apt-get install nginx
7
8   2)nginx 설정파일 수정하기
9     $ sudo nano /etc/nginx/nginx.conf
10      user ubuntu;          <---여기 수정
11      worker_processes auto;
12      pid /run/nginx.pid;
13      include /etc/nginx/modules-enabled/*.conf;
14
15      events {
16          worker_connections 768;
17          # multi_accept on;
18      }
19
20      http {
21
22          ##
23          # Basic Settings
24          ##
25
26 3)myproject.conf 파일 생성하기
27   - .config 디렉토리 밑에 nginx 디렉토리 생성
28     $ pwd
29       /home/ubuntu/DjangoHome/myproject/.config
30
31     $ mkdir nginx
32
33   - 그 밑에 myproject.conf 파일 생성하기
34
35     $ cd nginx
36     $ nano myproject.conf
37       server {
38         listen 80;
39         server_name *.compute.amazonaws.com;
40         charset utf-8;
41         client_max_body_size 128M;
42
43         location / {
44             uwsgi_pass unix:///tmp/myproject.sock;
45             include uwsgi_params;
46         }
47     }
48
49 4)위 설정의 설명은 다음과 같다.
50   -listen
51     --요청을 받을 포트 번호를 의미.
52     --nginx는 80 포트가 http 기본 포트로 인정한다.
53   -server_name
54     --요청을 받을 서버 주소를 의미
55     --AWS EC2 인스턴스의 [퍼블릭 IPv4 DNS] 값
56   -location / { }
57     --"server_name/" 식의 요청이 들어올 경우, 처리할 내용에 대해서 정의
58     --location /static/ { } 의 경우엔 "server_name/static/" 주소 요청이 올 경우이다.
59
60 5).config/uwsgi/myproject.ini 수정하기
61   $ nano myproject.ini
62     [uwsgi]
63     chdir = /home/ubuntu/DjangoHome/myproject
64     module = myproject.wsgi:application
65     home = /home/ubuntu/myvenv/
66
67     uid = ubuntu
68     gid = ubuntu
69
70     http = :8000      <---- 삭제할 것
71     socket = /tmp/myproject.sock  <--- 새로 추가
72     chmod-socket = 666      <--- 새로 추가
73     chown-socket = ubuntu:ubuntu  <--- 새로 추가
74
75     enable-threads = true
76     master = true
77     vacuum = true
78     pidfile = /tmp/myproject.pid
79     logto = /var/log/uwsgi/myproject/$(exec://date +%Y-%m-%d).log
80     log-reopen = true
81
82 6)uwsgi.service 파일 생성
83   -항상 Nginx가 실행되고 있어야 한다.
84   -또한 uwsgi 서비스도 실행되고 있어야 한다.

```

-uwsgi를 계속 커들 수 있도록 설정 파일을 추가해서 백그라운드에 계속 실행하도록 한다.  
-프로젝트 디렉토리의 .config 폴더의 uwsgi 폴더안에 uwsgi.service 파일을 만들고 다음과 같이 입력 후 저장한다.

```
$ nano uwsgi.service

[Unit]
Description=uWSGI service
After=syslog.target

[Service]
ExecStart=/home/ubuntu/myvenv/bin/uwsgi -i /home/ubuntu/DjangoHome/myproject/.config/uwsgi/myproject.ini

Restart=always
KillSignal=SIGQUIT
Type=notify
StandardError=syslog
NotifyAccess=all

[Install]
WantedBy=multi-user.target
```

7)uwsgi.service 파일 저장 후 데몬(백그라운드에 실행)에 등록한다.

-이 파일을 /etc/systemd/system/ 에 링크를 걸어준다.

```
$ sudo ln -f /home/ubuntu/DjangoHome/myproject/.config/uwsgi/uwsgi.service /etc/systemd/system/uwsgi.service
```

8)데몬을 새로고침 한다.

```
$ sudo systemctl daemon-reload
```

9)uwsgi 서비스를 사용 가능하게 변경해주고 restart 한 번 한다.

```
$ sudo systemctl enable uwsgi
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/uwsgi.service → /etc/systemd/system/uwsgi.service.
```

```
$ sudo systemctl restart uwsgi
```

10)Django 프로젝트 내의 nginx 설정 파일을 nginx 어플리케이션에 등록해 주어야 한다.

-cp 명령어를 이용해 등록하는 경로(sites-available)로 파일을 복사한다.

```
$ sudo cp -f /home/ubuntu/DjangoHome/myproject/.config/nginx/myproject.conf
/etc/nginx/sites-available/myproject.conf
```

11)sites-available에 복사된 설정 파일을 sites-enabled폴더 안에서도 링크한다.

```
$ sudo ln -sf /etc/nginx/sites-available/myproject.conf /etc/nginx/sites-enabled/myproject.conf
```

12)sites-enabled폴더 안의 default 파일을 삭제한다.

```
$ sudo rm /etc/nginx/sites-enabled/default
```

13)다시 데몬을 새로 고침 해주고 nginx와 uwsgi를 다시 실행한다.

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl restart uwsgi nginx
```

14)Web Browser에서 확인

```
http://{EC2 Instance 퍼블릭 IPv4 DNS값}/
ex)http://ec2-43-201-250-26.ap-northeast-2.compute.amazonaws.com/
```

15)지금까지의 수정된 내용을 Github에 Push한다.

-git 상태 파악

```
$ git status
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git restore <file>..." to discard changes in working directory)
```

```
modified: myproject.ini
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
../nginx/
```

```
uwsgi.service
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

-git add

```
$ git add .
```

-다시 git 상태 파악

```
$ git status
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Changes to be committed:
```

```
(use "git restore --staged <file>..." to unstage)
```

```
modified: myproject.ini
```

```
new file: uwsgi.service
```

```

168 Untracked files:
169 (use "git add <file>..." to include in what will be committed)
170 ../nginx/
171
172 -commit
173 $ git commit -m "Update configuration to connect Nginx"
174 [main 6568828] Update configuration to connect Nginx
175 2 files changed, 18 insertions(+), 1 deletion(-)
176 create mode 100644 myproject/.config/uwsgi/uwsgi.service
177
178 -push
179 $ git push origin main
180 Username for 'https://github.com': gitinstructor
181 Password for 'https://gitinstructor@github.com':
182 Enumerating objects: 12, done.
183 Counting objects: 100% (12/12), done.
184 Compressing objects: 100% (6/6), done.
185 Writing objects: 100% (7/7), 801 bytes | 801.00 KiB/s, done.
186 Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
187 remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
188 To https://github.com/gitinstructor/DjangoHome.git
189 6d34994..6568828 main -> main
190
191

```

## 2. static 파일 연결하기

```

193 1)현재 ~/admin/으로 방문하면 제대로 페이지가 보이지 않는다.
194 2)static 파일들이 제대로 로딩되지 않는다.
195 3)그 이유는 static 파일의 경로가 nginx에서 설정되어 있지 않기 때문이다.
196 4)static 파일들은 Django 프로젝트 내에 앱 별로 구분되어 놓여있는 경우가 대부분이기 때문에, 이 static 파일들을 하나의 경로로 먼저 모아둘 필요가 있다.
197 5)이를 쉽게 해주는 명령어는 python manage.py collectstatic이다.
198 6)이 명령어를 사용하려면 settings.py에 static 파일이 모이는 경로를 설정해 주어야 한다.
199 7)settings.py를 열고 os module을 import한다.

```

```

200 $ nano settings.py
201 ...
202 from pathlib import Path
203 import os          <----새로 추가
204

```

8)settings.py 파일의 제일 하단 STATIC\_URL 밑에 아래 값을 추가한다.

```

206 ...
207 # Static files (CSS, JavaScript, Images)
208 # https://docs.djangoproject.com/en/4.2/howto/static-files/
209
210 STATIC_URL = 'static/'
211 STATIC_ROOT = os.path.join(BASE_DIR, 'static')    <----새로 추가하기
212
213 # Default primary key field type
214 # https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field
215
216 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
217

```

9)STATIC\_ROOT 값을 추가한 뒤에 collectstatic 명령어를 사용하면 STATIC\_ROOT의 경로에 각 앱의 static 파일들이 모아진다.

10)위의 값은 프로젝트 폴더 안에 static 폴더를 만들어 그곳에 모이게 한다. (BASE\_DIR이 프로젝트 폴더의 절대 경로를 의미.)

11)collectstatic 명령어 사용

```

222 $ python manage.py collectstatic
223
224 125 static files copied to '/home/ubuntu/DjangoHome/myproject/static'.
225

```

```

226 $ ls
227 db.sqlite3  main  manage.py  myproject  requirements.txt  static
228

```

12)확인

```

230 -static > admin > css
231 -static > admin > img
232 -static > admin > js
233

```

13)myproject.conf로 설정한 Nginx는 현재 static 파일들이 어디에 있는지 알 수 없다.

-myproject.conf 파일에서 /static/ 요청시 파일의 경로를 입력한다.

```

237 $ nano .config/nginx/myproject.conf
238 server {
239     listen 80;
240     server_name *.compute.amazonaws.com;
241     charset utf-8;
242     client_max_body_size 128M;
243
244     location / {
245         uwsgi_pass unix:///tmp/myproject.sock;
246         include uwsgi_params;
247     }
248
249     # 아래 3줄 추가
250     location /static/ {
251         alias /home/ubuntu/DjangoHome/myproject/static/;

```

```

252     }
253 }
254
255 14)myproject.conf가 수정됐기 때문에 다시 Nginx에 등록해야 한다.
256 $ sudo cp -f /home/ubuntu/DjangoHome/myproject/.config/nginx/myproject.conf /etc/nginx/sites-available/myproject.conf
257 $ sudo ln -sf /etc/nginx/sites-available/myproject.conf /etc/nginx/sites-enabled/myproject.conf
258
259 15)Nginx와 uwsgi 서비스를 재시작한다.
260 $ sudo systemctl daemon-reload
261 $ sudo systemctl restart uwsgi nginx
262
263 16)Web Brower에서 확인
264
265 17)지금까지의 수정된 내용을 Github에 Push한다.
266 -git 상태 파악
267 $ git status
268 On branch main
269 Your branch is up to date with 'origin/main'.
270
271 Changes not staged for commit:
272 (use "git add <file>..." to update what will be committed)
273 (use "git restore <file>..." to discard changes in working directory)
274 modified:   ../../myproject/settings.py
275
276 Untracked files:
277 (use "git add <file>..." to include in what will be committed)
278 ../../.config/nginx/
279 ../../
280
281 no changes added to commit (use "git add" and/or "git commit -a")
282
283 -git add
284 $ git add .
285
286 -다시 git 상태 파악
287 $ git status
288 On branch main
289 Your branch is up to date with 'origin/main'.
290
291 Changes to be committed:
292 (use "git restore --staged <file>..." to unstage)
293 new file:   autocomplete.css
294 new file:   base.css
295 new file:   changelists.css
296 new file:   dark_mode.css
297 new file:   dashboard.css
298 new file:   forms.css
299 new file:   login.css
300 new file:   nav_sidebar.css
301 new file:   responsive.css
302 new file:   responsive_rtl.css
303 new file:   rtl.css
304 new file:   vendor/select2/LICENSE-SELECT2.md
305 new file:   vendor/select2/select2.css
306 new file:   vendor/select2/select2.min.css
307 new file:   widgets.css
308
309 Changes not staged for commit:
310 (use "git add <file>..." to update what will be committed)
311 (use "git restore <file>..." to discard changes in working directory)
312 modified:   ../../myproject/settings.py
313
314 Untracked files:
315 (use "git add <file>..." to include in what will be committed)
316 ../../.config/nginx/
317 ../img/
318 ../js/
319
320 -commit
321 $ git commit -m "Update configuration to use static files"
322 [main 124553e] Update configuration to use static files
323 15 files changed, 5111 insertions(+)
324 create mode 100755 myproject/static/admin/css/autocomplete.css
325 create mode 100755 myproject/static/admin/css/base.css
326 create mode 100755 myproject/static/admin/css/changelists.css
327 create mode 100755 myproject/static/admin/css/dark_mode.css
328 create mode 100755 myproject/static/admin/css/dashboard.css
329 create mode 100755 myproject/static/admin/css/forms.css
330 create mode 100755 myproject/static/admin/css/login.css
331 create mode 100755 myproject/static/admin/css/nav_sidebar.css
332 create mode 100755 myproject/static/admin/css/responsive.css
333 create mode 100755 myproject/static/admin/css/responsive_rtl.css
334 create mode 100755 myproject/static/admin/css/rtl.css
335 create mode 100755 myproject/static/admin/css/vendor/select2/LICENSE-SELECT2.md

```

```
336 create mode 100755 myproject/static/admin/css/vendor/select2/select2.css
337 create mode 100755 myproject/static/admin/css/vendor/select2/select2.min.css
338 create mode 100755 myproject/static/admin/css/widgets.css
339
340 -push
341 $ git push origin main
342 Username for 'https://github.com': gitinstructor
343 Password for 'https://gitinstructor@github.com':
344 Enumerating objects: 25, done.
345 Counting objects: 100% (25/25), done.
346 Compressing objects: 100% (20/20), done.
347 Writing objects: 100% (23/23), 25.04 KiB | 3.58 MiB/s, done.
348 Total 23 (delta 2), reused 0 (delta 0), pack-reused 0
349 remote: Resolving deltas: 100% (2/2), completed with 1 local object.
350 To https://github.com/gitinstructor/DjangoHome.git
351 6568828..124553e main -> main
352
```