

# Django via docker®

Deploying Django Apps with Docker

Autumn 2024



AI융합학과

Seongbok Baik

sbbaik@dju.ac.kr



장철원 소프트웨어공학자

# 01 Django Project 생성

Django Project 생성에 필요한 폴더 준비

```
eevee@myserver01:~$ ls 1
work
eevee@myserver01:~$ cd work/ 2
eevee@myserver01:~/work$ ls 3
ch04 ch05
eevee@myserver01:~/work$ cd ch05 4
eevee@myserver01:~/work/ch05$ ls 5
ex01
eevee@myserver01:~/work/ch05$ mkdir ex02 6
eevee@myserver01:~/work/ch05$ cd ex02 7
eevee@myserver01:~/work/ch05/ex02$
```

## 02 Django Project 생성

### Django Project 생성

```
eevee@myserver01:~/work/ch05/ex02$ pyenv activate py3_11_6
(py3_11_6) eevee@myserver01:~/work/ch05/ex02$ django-admin startproject myapp
(py3_11_6) eevee@myserver01:~/work/ch05/ex02$ ls
myapp
```

1

2

3

### Django Project Directory 확인

```
(py3_11_6) eevee@myserver01:~/work/ch05/ex02$ tree ./
```

4

```
./
├── myapp
│   ├── manage.py
│   └── myapp
│       ├── asgi.py
│       ├── __init__.py
│       ├── settings.py
│       ├── urls.py
│       └── wsgi.py
2 directories, 6 files
```

## 03 Django Project 생성

Django Project 실행 환경 설정

```
(py3_11_6) eevee@myserver01:~/work/ch05/ex02$ cd myapp/myapp/  
(py3_11_6) eevee@myserver01:~/work/ch05/ex02/myapp/myapp$ vim settings.py
```

1

생성 중인 Django Project으로 접근할 수 있는 외부 범위 설정

```
# SECURITY WARNING: don't  
DEBUG = True  
  
ALLOWED_HOSTS = []
```

before



```
# SECURITY WARNING: don't  
DEBUG = True  
  
ALLOWED_HOSTS = [ '*' ]
```

after

ALLOWED\_HOSTS = [] 에서 []를 ['\*'] (<= all, 모든 호스트 접속 가능) 로 변경

## 04 Django Project 생성

migrate 명령으로 현재 Django Project를 실제 데이터베이스에 반영

```
(py3_11_6) eevee@myserver01:~/work/ch05/ex02/myapp/myapp$ cd .. ①
(py3_11_6) eevee@myserver01:~/work/ch05/ex02/myapp$ ls ②
manage.py  myapp
(py3_11_6) eevee@myserver01:~/work/ch05/ex02/myapp$ python manage.py migrate ③
```

### Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

### Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

...(중략)

Applying auth.0011\_update\_proxy\_permissions... OK

Applying auth.0012\_alter\_user\_first\_name\_max\_length... OK

Applying sessions.0001\_initial... OK

## 05 Django Project 생성

runserver 명령으로 현재 Django Project를 실행

```
(py3_11_6) eevee@myserver01:~/work/ch05/ex02/myapp$ python manage.py runserver  
0.0.0.0:8000
```

1

```
Watching for file changes with StatReloader  
Performing system checks...
```

```
System check identified no issues (0 silenced).  
November 02, 2023 - 11:04:21  
Django version 4.2.7, using settings 'myapp.settings'  
Starting development server at http://0.0.0.0:8000/  
Quit the server with CONTROL-C.
```

0.0.0.0:8000 => 8000번 포트를 사용하여 Django 서비스 제공

## 06 Django Project 생성

내용 수정: ~/work/ch05/ex02/myapp/myapp/urls.py 수정

```
from django.http import HttpResponse
from django.contrib import admin
from django.urls import path

def home(request):
    return HttpResponse("<h1>Welcome to Seongbok's Django Project!</h1>")

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', home), # Define the root route directly here
]
```

파일 수정 후 ~/work/ch05/ex02/myapp/에서 아래 명령 다시 실행

```
python manage.py migrate
python manage.py runserver 0.0.0.0:8000
```



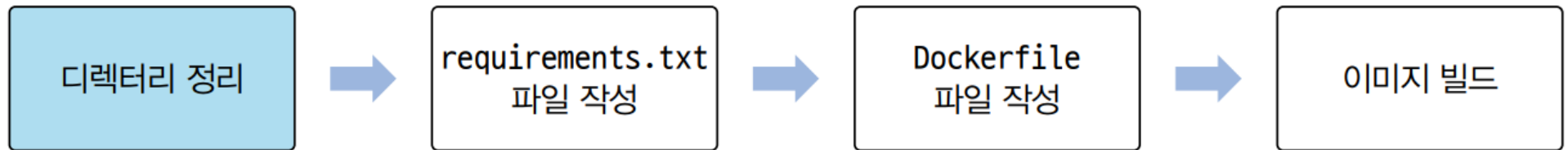
## 07 Django Project 생성

웹 브라우저로 <http://127.0.0.1:8000> 접속 실패 시 Firewall 점검

```
sudo ufw status  
sudo ufw allow 8000/tcp  
sudo ufw reload
```

## 08 Django Image Building

Django Image Building 과정



## 09 Django Image Building

Django Image Building을 위한 디렉토리 준비

```
eevee@myserver01:~$ cd work/ch05/
```

1

```
eevee@myserver01:~/work/ch05$ ls
```

2

```
ex01  ex02
```

```
eevee@myserver01:~/work/ch05$ cp -r ex02 ex03
```

3

```
eevee@myserver01:~/work/ch05$ ls
```

4

```
ex01  ex02  ex03
```

## 10 Django Image Building

Django Image Building을 위한 디렉토리 확인

```
eevee@myserver01:~/work/ch05$ cd ex03 5
eevee@myserver01:~/work/ch05/ex03$ tree ./ -L 3 6
./
├── myapp
│   ├── db.sqlite3
│   ├── manage.py
│   └── myapp
│       ├── asgi.py
│       ├── __init__.py
│       ├── __pycache__
│       ├── settings.py
│       ├── urls.py
│       └── wsgi.py
3 directories, 7 files
```

## 11 Django Image Building

Django Image Building을 위해 필요한 설치 파일 목록인 requirements.txt 파일 생성

```
eevee@myserver01:~/work/ch05/ex03$ ls 1
myapp
eevee@myserver01:~/work/ch05/ex03$ vim requirements.txt 2
eevee@myserver01:~/work/ch05/ex03$ cat requirements.txt 3
django==4.2.7
eevee@myserver01:~/work/ch05/ex03$ ls 4
myapp requirements.txt
```

requirements.txt 파일에는 현재 설치된 django 패키지의 버전 명시

## 12 Django Image Building

도커 Image 파일 생성에 필요한 Dockerfile 준비

```
eevee@myserver01:~/work/ch05/ex03$ vim Dockerfile 1
FROM python:3.11.6 2

WORKDIR /usr/src/app 3

COPY . . 4

RUN python -m pip install --upgrade pip 5
RUN pip install -r requirements.txt 6

WORKDIR ./myapp 7
CMD python manage.py runserver 0.0.0.0:8000 8
EXPOSE 8000 9

eevee@myserver01:~/work/ch05/ex03$ ls 10
Dockerfile myapp requirements.txt
```

## 13 Django Image Building

도커 이미지 생성

```
eevee@myserver01:~/work/ch05/ex03$ docker image build . -t myweb01
[+] Building 6.5s (11/11) FINISHED
=> [internal] load.dockerignore 0.1s
=> => transferring context: 2B
...(중략)
=> [6/6] WORKDIR ./myapp
=> exporting to image
=> => exporting layers
=> => writing image sha256:c8ba32ed60bc4958ddd03281811cc864670d506e40d6c8f921b6
=> => naming to docker.io/library/myweb01
```

"build ." => 현재 디렉토리를 이미지로 빌드

## 14 Django Image Building

생성된 도커 이미지 확인

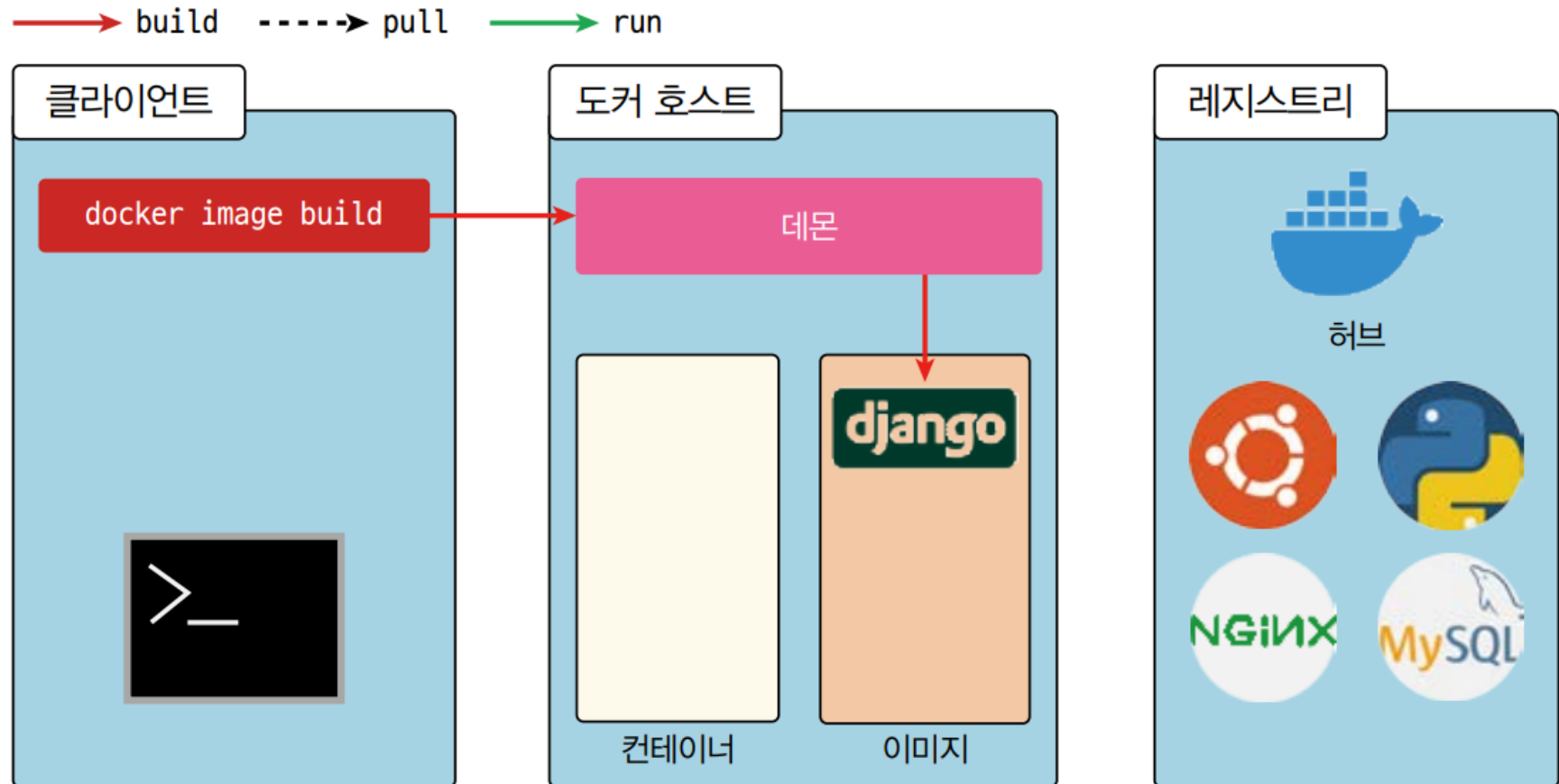
```
eevee@myserver01:~/work/ch05/ex03$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
myweb01	latest	c8ba32ed60bc	2 minutes ago	1.07GB

1



# 15 Django Image Building



## 16 Django 컨테이너 실행

생성된 도커 이미지를 컨테이너 형태로 실행

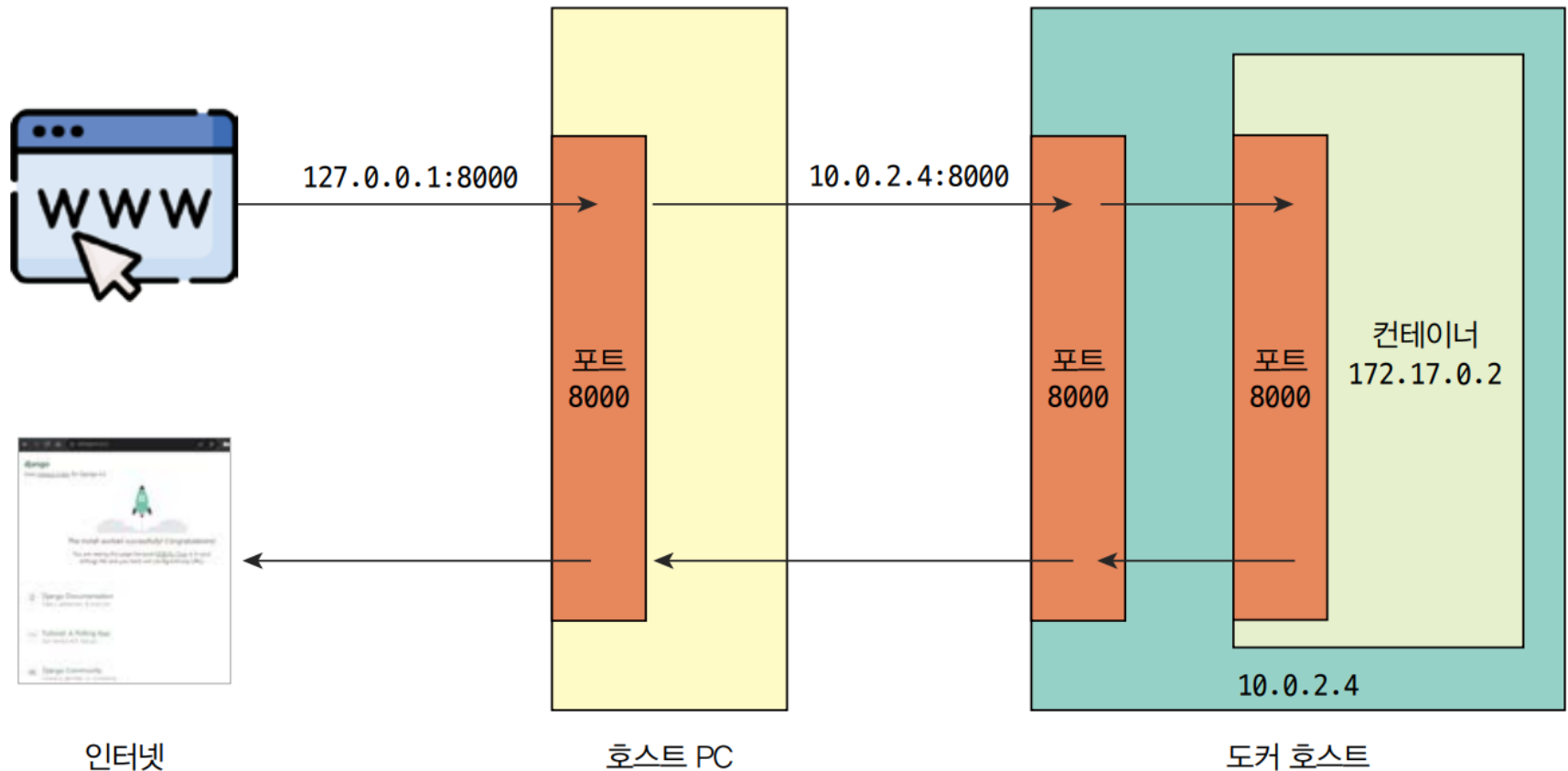
```
eevee@myserver01:~/work/ch05/ex03$ docker container run -d -p 8000:8000 myweb01 1  
8f3b263b41f8fe463bc26b696563b6f0394c08e2d4a82fdfa6f79daa42b39ba7
```

```
eevee@myserver01:~/work/ch05/ex03$ docker container ls 2
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
8f3b263b41f8	myweb01	"/bin/sh -c 'python ..."	6 seconds ago	Up
5 seconds	0.0.0.0:8000->8000/tcp, :::8000->8000/tcp	stoic_mayer		

- p : 포트 포워딩 옵션
- d : 백그라운드 실행 지정

# 17 컨테이너를 활용한 Django 서비스 실행 개념



서로에게, 자신에게 친절합니다 - 허준이

