```
Lab11. Docker Compose를 사용하는 이유
1
2
   1. Docker 실행 명령어를 일일이 입력하기가 복잡해서
3
      1)Nginx 실행하기
4
5
         $ docker run -it nginx
         <--- 실행은 되지만 port binding되지 않아 브라우저로 연결 안됨. 바로 shell로 연결됨. Ctrl + C를 누르면 바로
6
         Container 종료됨. 외부에서 접근할 방법이 없다.
7
         $ curl localhost:80
         curl: (7) Failed to connect to localhost port 80 after 0 ms: Connection refused
8
9
10
      2)Nginx Container 실행 + Host 8080 Port 연결하기
         $ docker run -it -p 8080:80 nginx
11
         <--- 실행도 되고 웹브라우저로 연결 가능하지만, 바로 shell로 연결됨, Ctrl + C를 누르면 바로 Container 종료됨.
12
13
         $ curl localhost:8080
         <!DOCTYPE html>
14
15
         <html>
16
         <head>
17
         <title>Welcome to nginx!</title>
18
         <style>
19
         html { color-scheme: light dark; }
20
         body { width: 35em; margin: 0 auto;
         font-family: Tahoma, Verdana, Arial, sans-serif; }
21
22
         </style>
23
         </head>
24
         <body>
25
         <h1>Welcome to nginx!</h1>
         If you see this page, the nginx web server is successfully installed and
26
27
         working. Further configuration is required.
28
29
         For online documentation and support please refer to
         <a href="http://nginx.org/">nginx.org</a>.<br/>
30
         Commercial support is available at
31
         <a href="http://nginx.com/">nginx.com</a>.
32
33
34
         Thank you for using nginx.
35
         </body>
         </html>
36
37
38
      3)Nginx Container 실행 + Host 8080 Port 연결 + Container 종료시 자동 삭제
39
         $ docker run -it -p 8080:80 --rm nginx
         <---실행도 되고 웹브라우저로 연결 가능하지만, 바로 shell로 연결됨, Ctrl + C로 bash에서 나오면 바로
40
         Container 종료됨
41
      4)Nginx 컨테이너 실행 + Host 8080 Port 연결 + Container 종료시 자동 삭제 + Host의 Directory를 Container
42
      안에서 링크하기
43
         $ vi index.html
         <h1>Hello, Docker Compose World!!!</h1>
44
45
         $ docker run -it -p 8080:80 --rm -v $PWD:/usr/share/nginx/html nginx
46
         <---실행은 되지만, 웹브라우저로 접속시 403 (13: Permission denied) Error 발생
47
         <---오류를 해결하려면 nginx Container의 /etc/nginx/nginx.conf파일 첫 줄에 user root;를 넣어야 함.
48
49
         <---어쨌든 복잡함.
         $ curl localhost:8080
50
51
         <html>
52
         <head><title>403 Forbidden</title></head>
53
         <body>
54
         <center><h1>403 Forbidden</h1></center>
55
         <hr> <center>nginx/1.25.3</center>
56
         </body>
         </html>
57
58
```

```
59
    2. 컨테이너끼리 연결하기 편해서 --1) 후 바로 --3)실행
 60
       1)준비 : django-sample 이미지를 빌드
 61
 62
          $ git clone https://github.com/raccoonyy/django-sample-for-docker-compose.git django-sample
          $ cd django-sample
 63
 64
          $ docker build -t django-sample .
 65
 66
       2)django 컨테이너 실행 + postgres 컨테이너 실행
          $ docker run --rm -d --name django -p 8000:8000 django-sample
 67
 68
 69
          $ docker ps -a
 70
 71
          -Web Browser를 열고 http://ip:8000
            --django 잘 실행되고 있음을 확인
 72
 73
 74
          $ docker run --rm -d --name postgres -e POSTGRES_DB=djangosample ₩
 75
          > -e POSTGRES_USER=sampleuser ₩
 76
          > -e POSTGRES_PASSWORD=samplesecret ₩
 77
          > postares
 78
 79
          -Web Browser를 열고 http://ip:8000
            --그냥 django만 잘 실행되고 있음.
 80
 81
          -다음 3) 실행을 위해 모든 Docker Image와 Container 삭제하기
 82
 83
            $ docker rm -f `docker ps -a -q`
            $ docker rmi -f `docker images -q`
 84
 85
          -django-sample image 다시 build
 86
            $ docker build -t django-sample .
 87
 88
       3)postgres 컨테이너 실행 + diango 컨테이너 실행 + 서로 연결하기
 89
          $ docker run --rm -d --name postgres -e POSTGRES_DB=djangosample ₩
 90
 91
          > -e POSTGRES USER=sampleuser ₩
 92
          > -e POSTGRES_PASSWORD=samplesecret ₩
 93
          > postgres
 94
 95
          $ docker run -d --rm -p 8000:8000 -e DJANGO_DB_HOST=db ₩
          > --link postgres:db ₩
 96
          > django-sample
 97
 98
 99
100
    3. 특정 컨테이너끼리만 통신할 수 있는 가상 네트워크 환경을 편리하게 관리하고 싶어서
       1)postgres 컨테이너 실행 + django1 컨테이너 연결
101
102
          $ docker run --rm -d --name postgres ₩
          > -e POSTGRES_DB=djangosample ₩
103
          > -e POSTGRES_USER=sampleuser ₩
104
          > -e POSTGRES_PASSWORD=samplesecret ₩
105
          > postgres
106
107
108
          $ docker run -d --rm --name django1 ₩
          > -p 8000:8000 ₩
109
110
          > -e DJANGO_DB_HOST=db ₩
111
          > --link postgres:db ₩
          > django-sample
112
113
       2)postgres 컨테이너는 호스트의 다른 컨테이너들이 모두 접근할 수 있음
114
          $ docker run -d --rm --name django2 ₩
115
          > -p 8001:8000 ₩
116
          > -e DJANGO DB HOST=db ₩
117
118
          > --link postgres:db ₩
119
          > django-sample
```

```
120
        3)postgres 컨테이너 + django1 컨테이너만 통신할 수 있는 가상 네트워크 만들기
121
          -도커 네트워크 살펴보기
122
123
             $ docker network Is
124
          -도커 네트워크 생성하기
125
             $ docker network create --driver bridge web-service
126
127
             $ docker network Is
128
          -컨테이너 실행하기
129
             $ docker rm -f postgres
130
             $ docker run --rm -d --name postgres ₩
131
132
             > --network web-service ₩
             > -e POSTGRES_DB=djangosample ₩
133
134
             > -e POSTGRES_USER=sampleuser ₩
135
             > -e POSTGRES_PASSWORD=samplesecret ₩
             > postgres
136
137
138
             $ docker rm -f django1
             $ docker run -d --rm --name django1 ₩
139
             > --network web-service ₩
140
             > -p 8000:8000 ₩
141
142
             > -e DJANGO_DB_HOST=db ₩
             > --link postgres:db ₩
143
144
             > django-sample
145
             $ docker rm -f django2
146
             $ docker run -d --rm --name django2 ₩
147
             > -p 8001:8000 ₩
148
             > -e DJANGO_DB_HOST=db ₩
149
             > --link postgres:db ₩
150
151
             > django-sample
152
             docker: Error response from daemon: Cannot link to /postgres, as it does not belong to the default
             network.
153
154
155
156 4. 이 모든 것을 간단한 명령어로 관리하고 싶어서
157
        -준비
158
          $ docker rm -f `docker container ps -a -q`
          $ dockre rmi -f `docker image Is -q`
159
160
          $ docker build -t django-sample .
          $ docker network rm web-service
161
162
        1)실행 명령어와 종료 명령어
163
          $ docker network create --driver bridge web-service
164
165
166
          $ docker run --rm -d --name postgres ₩
167
          > --network web-service ₩
168
          > -p 5432:5432 ₩
          > -e POSTGRES_DB=djangosample ₩
169
170
          > -e POSTGRES_USER=sampleuser ₩
171
          > -e POSTGRES PASSWORD=samplesecret ₩
172
          > postgres
173
          $ docker run -d --rm --name django1 ₩
174
          > --network web-service ₩
175
          > -p 8000:8000 ₩
176
          > -e DJANGO_DB_HOST=db ₩
177
178
          > --link postgres:db ₩
179
          > django-sample
```

```
180
181
          $ docker kill django1 postgres
          $ docker network rm web-service
182
183
184
        2)docker-compose.yml 생성하기
185
          $ cd django-sample
186
187
          $ nano docker-compose.yml
188
          version: '3'
189
190
191
          volumes:
192
             postgres_data: {}
193
          services:
194
195
             db:
196
                image: postgres
197
                volumes:
198
                 - postgres_data:/var/lib/postgres/data <--- ':'과 '/'는 반드시 붙인다.
199
                environment:
200
                 - POSTGRES_DB=djangosample
                 - POSTGRES_USER=sampleuser
201
202
                 - POSTGRES_PASSWORD=samplesecret
203
             django:
204
                build:
205
                   context: .
                   dockerfile: ./compose/django/Dockerfile-dev
206
207
                   volumes:
208
                    - ./:/app/
                   command: ["./manage.py", "runserver", "0:8000"]
209
210
                   environment:
                    - DJANGO_DB_HOST=db
211
212
                   depends on:
                    - db
213
214
                   restart: always
215
                   ports:
216
                    - 8000:8000
217
218
          3)도커 컴포즈로 실행하고 종료하기
219
220
             -모든 docker process 중지
221
             -모든 docker images 삭제
             -docker network도 기본 3개만 남기고 모두 삭제
222
223
                                            <---반드시 django-sample directory 안에서 실행할 것
224
             $ docker-compose up -d
225
             $ docker-compose down
226
             -웹 브라우저로 확인할 것
227
```