

```

1 1. Docker 실행 명령어를 일일이 입력하기가 복잡해서
2 1)Nginx 실행하기
3   $ sudo docker run -it nginx
4   <--- 실행은 되지만 웹브라우저로 연결 불능, 바로 shell로 연결됨.
5
6 2)Nginx Container 실행 + Host 8080 Port 연결하기
7   $ sudo docker run -it -p 8080:80 nginx
8   <---실행도 되고 웹브라우저로 연결 가능하지만, 바로 shell로 연결됨.
9
10 3)Nginx Container 실행 + Host 8080 Port 연결 + Container 종료시 자동 삭제
11   $ sudo docker run -it -p 8080:80 --rm nginx
12   <---실행도 되고 웹브라우저로 연결 가능하지만, 바로 shell로 연결됨, Ctrl + C로 bash에 나오면 바로 Container 종료됨.
13
14 4)Nginx 컨테이너 실행 + Host 8080 Port 연결 + Container 종료시 자동 삭제 + Host의 Directory를 Container 안에서 링크하기
15   $ vi index.html
16   <h1>Hello, Docker Compose World!!!</h1>
17
18   $ sudo docker run -it -p 8080:80 --rm -v ${PWD}:/usr/share/nginx/html/ nginx
19   <---실행은 되지만, 웹브라우저로 접속시 403 (13: Permission denied) Error 발생
20   <---오류를 해결하려면 nginx Container의 /etc/nginx/nginx.conf파일 첫 줄에 user root;를 넣어야 함.
21   <---어쨌든 복잡함.
22
23
24 2. 컨테이너끼리 연결하기 편해서 --1) 후 바로 --3)실행
25 1)준비 : django-sample 이미지를 빌드
26   $ git clone https://github.com/raccoonvy/django-sample-for-docker-compose.git django-sample
27   $ cd django-sample
28   $ docker build -t django-sample .
29
30 2)django 컨테이너 실행 + postgres 컨테이너 실행
31   $ docker run --rm -d --name django -p 8000:8000 django-sample
32
33   $ docker ps -a
34
35   -Web Browser를 열고 http://ip:8000
36   --django 잘 실행되고 있음을 확인
37
38   $ docker run --rm -d --name postgres -e POSTGRES_DB=djangosample \
39   > -e POSTGRES_USER=sampleuser \
40   > -e POSTGRES_PASSWORD=samplesecret \
41   > postgres
42
43   -Web Browser를 열고 http://ip:8000
44   --그냥 django만 잘 실행되고 있음.
45
46   -다음 3) 실행을 위해 모든 Docker Image와 Container 삭제하기
47   $ docker rm -f `docker ps -a -q`
48   $ docker rmi -f `docker images -q`
49
50   -django-sample image 다시 build
51   $ docker build -t django-sample .
52
53 3)postgres 컨테이너 실행 + django 컨테이너 실행 + 서로 연결하기
54   $ docker run --rm -d --name postgres -e POSTGRES_DB=djangosample \
55   > -e POSTGRES_USER=sampleuser \
56   > -e POSTGRES_PASSWORD=samplesecret \
57   > postgres
58
59   $ docker run -d --rm -p 8000:8000 -e DJANGO_DB_HOST=db \
60   > --link postgres:db \
61   > django-sample
62
63
64 3. 특정 컨테이너끼리만 통신할 수 있는 가상 네트워크 환경을 편리하게 관리하고 싶어서
65 1)postgres 컨테이너 실행 + django1 컨테이너 연결
66   $ docker run --rm -d --name postgres \
67   > -e POSTGRES_DB=djangosample \
68   > -e POSTGRES_USER=sampleuser \
69   > -e POSTGRES_PASSWORD=samplesecret \
70   > postgres
71
72   $ docker run -d --rm --name django1 \
73   > -p 8000:8000 \
74   > -e DJANGO_DB_HOST=db \
75   > --link postgres:db \
76   > django-sample
77
78 2)postgres 컨테이너는 호스트의 다른 컨테이너들이 모두 접근할 수 있음
79   $ docker run -d --rm --name django2 \
80   > -p 8001:8000 \
81   > -e DJANGO_DB_HOST=db \
82   > --link postgres:db \
83   > django-sample
84

```

```

85 3)postgres 컨테이너 + django1 컨테이너만 통신할 수 있는 가상 네트워크 만들기
86   -도커 네트워크 살펴보기
87     $docker network ls
88
89   -도커 네트워크 생성하기
90     $ docker network create --driver bridge web-service
91     $ docker network ls
92
93   -컨테이너 실행하기
94     $ docker run --rm -d --name postgres \
95       > --network web-service \
96       > -e POSTGRES_DB=djangosample \
97       > -e POSTGRES_USER=sampleuser \
98       > -e POSTGRES_PASSWORD=samplesecret \
99       > postgres
100
101     $ docker run -d --rm --name django1 \
102       > --network web-service \
103       > -p 8000:8000 \
104       > -e DJANGO_DB_HOST=db \
105       > --link postgres:db \
106       > django-sample
107
108     $ docker run -d --rm --name django2 \
109       > -p 8001:8000 \
110       > -e DJANGO_DB_HOST=db \
111       > --link postgres:db \
112       > django-sample
113     docker: Error response from daemon: Cannot link to /postgres, as it does not belong to the default network.
114
115
116

```

#### 4. 이 모든 것을 간단한 명령어로 관리하고 싶어서

```

117 1)실행 명령어와 종료 명령어
118   $ docker network create --driver bridge web-service
119
120
121   $ docker run --rm -d --name postgres \
122     > --network web-service \
123     > -p 5432:5432 \
124     > -e POSTGRES_DB=djangosample \
125     > -e POSTGRES_USER=sampleuser \
126     > -e POSTGRES_PASSWORD=samplesecret \
127     > postgres
128
129   $ docker run -d --rm --name django1 \
130     > --network web-service \
131     > -p 8000:8000 \
132     > -e DJANGO_DB_HOST=db \
133     > --link postgres:db \
134     > django-sample
135
136   $ docker kill django1 postgres
137   $ docker network rm web-service
138
139
140 2)docker-compose.yml 생성하기
141   $ cd django-sample
142   $ nano docker-compose.yml
143
144   version: '3'
145
146   volumes:
147     postgres_data: {}
148
149   services:
150     db:
151       image: postgres
152       volumes:
153         - postgres_data:/var/lib/postgres/data <--- ':'과 '/'는 반드시 붙인다.
154       environment:
155         - POSTGRES_DB=djangosample
156         - POSTGRES_USER=sampleuser
157         - POSTGRES_PASSWORD=samplesecret
158     django:
159       build:
160         context: .
161         dockerfile: ./compose/django/Dockerfile-dev
162       volumes:
163         - ./:/app/
164       command: ["/manage.py", "runserver", "0:8000"]
165       environment:
166         - DJANGO_DB_HOST=db
167       depends_on:
168         - db

```

```
169         restart: always
170         ports:
171             - 8000:8000
172
173 3)도커 컴포즈로 실행하고 종료하기
174     -모든 docker process 중지
175     -모든 docker images 삭제
176     -docker network도 기본 3개만 남기고 모두 삭제
177
178     $ docker-compose up -d          <---반드시 django-sample directory 안에서 실행할 것
179     $ docker-compose down
180
181     -웹 브라우저로 확인할 것
```