

```
1 Lab. Docker Compose Lab
2
3 1. Flask App을 Docker Compose로 실행하기
4   1)Flask Container
5     -Connection Port : 5000
6     -Redis Host Name : redis
7
8   2)Redis Container
9     -Image : redis
10
11  3)app.py
12    import time
13
14    import redis
15    from flask import Flask
16
17    app = Flask(__name__)
18    cache = redis.Redis(host='redis', port=6379)
19
20
21    def get_hit_count():
22        retries = 5
23        while True:
24            try:
25                return cache.incr('hits')
26            except redis.exceptions.ConnectionError as exc:
27                if retries == 0:
28                    raise exc
29                retries -= 1
30                time.sleep(0.5)
31
32
33    @app.route('/')
34    def hello():
35        count = get_hit_count()
36        return 'Hello World! I have been seen {} times.\n'.format(count)
37
38  4)requirements.txt
39    flask
40    redis
41
42  5)Dockerfile
43    FROM      python:3.7-alpine
44    WORKDIR   /code
45    ENV       FLASK_APP app.py
46    ENV       FLASK_RUN_HOST 0.0.0.0
47    RUN       apk add --no-cache gcc musl-dev linux-headers
48    COPY      requirements.txt requirements.txt
49    RUN       pip install -r requirements.txt
50    COPY      . .
51    CMD       ["flask", "run"]
52
53  6)확인 순서
54    -flask Application을 Build하여 Image를 생성
55    -50000 Port로 접속할 수 있게 docker-compose.yml 작성
56    -Docker Compose를 실행
57
58
59  7)Code
60    $ mkdir demo
61    $ cd demo
62    $ vim app.py
63    $ vim requirements.txt
64    $ vim Dockerfile
65
66    $ docker build -t flask-redis .
67
68    $ vim docker-compose.yml
69      version: '3'
70
71      services:
72        flask:
73          image: flask-redis
74          ports:
75            - 50000:5000
76        redis:
77          image: redis
78
79    $ docker-compose up
80
81    -Web Browser에서 확인
82    -http:{IP}:50000
83
84
```

```
85
86 2. Front-end, Back-end, Database로 구성된 방명록 서비스 실행하기
87 1)Front-end
88   -Image : subicura/guestbook-frontend:latest
89   -Port : 60000
90   -PORT 환경변수 : Service를 실행할 Port
91   -GUESTBOOK_API_ADDR 환경변수 : Back-end Server 주소 ex)backend:8000
92
93 2)Back-end
94   -Image : subicura/guestbook-backend:latest
95   -PORT 환경변수 : Service를 실행할 Port
96   -GUESTBOOK_DB_ADDR 환경변수 : Database Server 주소 ex)mongodb:27017
97
98 3)Database
99   -Image : mongo:4
100   -연결되는 Port : 27017
101   -Volume 설정 : /data/db
102
103
104 4)Code
105 $ mkdir demo
106 $ cd demo
107 $ vim docker-compose.yml
108
109     version: '3'
110
111     services:
112       frontend:
113         image: subicura/guestbook-frontend:latest
114         ports:
115           - 60000:3000
116         environment:
117           - PORT=3000
118           - GUESTBOOK_API_ADDR=backend:5000
119         depends_on:
120           - backend
121       backend:
122         image: subicura/guestbook-backend:latest
123         environment:
124           - PORT=5000
125           - GUESTBOOK_DB_ADDR=mongodb:27017
126         depends_on:
127           - mongodb
128
129       mongodb:
130         image: mongo:4
131         volumes:
132           - db_data:/data/db <---띄우지 말것
133
134     volumes:
135       db_data: {}
136
137 $ docker-compose up
138
139 -Web Browser에서
140   -http://{IP}:60000
```