

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

```

68     version: '3'
69
70     services:
71         flask:
72             image: flask-redis
73             ports:
74                 - 50000:5000
75             redis:
76                 image: redis
77
78 $ docker-compose up
79
80 -Web Browser에서 확인
81 -http:{IP}:50000
82
83
84
85 2. Front-end, Back-end, Database로 구성된 방명록 서비스 실행하기
86 1)Front-end
87     -Image : subicura/guestbook-frontend:latest
88     -Port : 60000
89     -PORT 환경변수 : Service를 실행할 Port
90     -GUESTBOOK_API_ADDR 환경변수 : Back-end Server 주소 ex)backend:8000
91
92 2)Back-end
93     -Image : subicura/guestbook-backend:latest
94     -PORT 환경변수 : Service를 실행할 Port
95     -GUESTBOOK_DB_ADDR 환경변수 : Database Server 주소 ex)mongodb:27017
96
97 3)Database
98     -Image : mongo:4
99     -연결되는 Port : 27017
100    -Volume 설정 : /data/db
101
102
103 4)Code
104     $ mkdir demo
105     $ cd demo
106     $ vim docker-compose.yml
107
108     version: '3'
109
110     services:
111         frontend:
112             image: subicura/guestbook-frontend:latest
113             ports:
114                 - 60000: 3000
115             environment:
116                 - PORT=3000
117                 - GUESTBOOK_API_ADDR=backend:5000
118             depends_on:
119                 - backend
120         backend:
121             image: subicura/guestbook-backend:latest
122             environment:
123                 - PORT=5000
124                 - GUESTBOOK_DB_ADDR=mongodb:27017
125             depends_on:
126                 - mongodb
127
128         mongodb:
129             image: mongo:4
130             volumes:
131                 - db_data:/data/db <---띄우지 말것
132
133     volumes:
134         db_data: {}

```

```
135
136 $ docker-compose up
137
138 -Web Browser에서
139   -http://{IP}:60000
```