

```
1 1. docker-compose pull
2   -pull 뒤에 이미지를 적지 않으면 모든 이미지를 다운로드한다.
3   1)$ cat docker-compose.yml
4       version: '3'
5       services:
6         nginx:
7           image: nginx:latest
8           ports:
9             - 8000:80
10
11   2)$ docker-compose pull
12
13
14 2. docker-compose build
15   -build 뒤에 이미지를 적지 않으면 필요한 이미지를 모두 빌드한다.
16   1)$ cat docker-compose.yml <-- 파일 수정
17       version: '3'
18       services:
19         nginx:
20           build:
21             context: . <---context가 현재 위치를 의미하고 있음. 같은 위치에 있는 Dockerfile을 사용할
                것임.
22             ports:
23               - 8000:80
24
25   2)Dockerfile
26       FROM nginx:latest
27
28       ADD ./index.html /usr/share/nginx/html/
29
30   3)index.html
31       <font size='7' color='blue'>Hello, Docker Compose</font>
32
33   4)$ docker-compose build
34
35
36 3. docker-compose up
37   -pull과 build를 실행했기 때문에 서비스를 구동한다.
38   1)$ docker-compose up
39       -브라우저에서 확인
40       --Ctrl + C로 중지
41
42   2)$ docker-compose up --build
43       -필요한 이미지를 강제로 빌드
44
45   3)$ docker-compose up --force-recreate
46       -이미 실행중인 컨테이너에 뭔가 문제가 생겼거나 docker-compose file을 수정했다면 컨테이너를
        새로 생성하면 된다.
47
48   4)$ docker-compose up -d
49       -서비스를 background로 실행, 실행 후 다시 터미널로 돌아옴.
50
51
52 4. docker-compose ps
53   1)서비스가 잘 실행되고 있는지 확인
54   2)Name의 앞부분은 현재 실행되고 있는 디렉토리
55       $ docker-compose ps
56
57
58 5. docker-compose logs
```

```
59 1)컨테이너별 로그를 보고 싶을 때
60     -$ docker-compose logs
61
62     -$ docker-compose logs -f
63     --브라우저에서 Refresh로 log가 계속 update되는 것을 확인
64
65 6. docker-compose top
66     $ docker-compose top
67     -지금 실행하고 있는 nginx의 Process를 확인
68
69
70 7. docker-compose stop
71     $ docker-compose stop
72     -stop뒤에 서비스 이름을 넣지 않으면 모든 서비스 중단
73
74     $ docker-compose ps
75     -중단된 서비스 확인
76     -State가 Exit됨을 확인
77
78
79 8. docker-compose start
80     -멈춘 서비스 다시 실행
81     $ docker-compose start
82
83
84 9. docker-compose exec
85     1)서비스 안에서 어떤 명령을 실행
86     -$ docker-compose exec nginx bash
87     /# touch TEST <---run과의 명령어 차이를 구별하기 위해 빈 파일 하나 생성
88     /# ls <---TEST 파일 확인
89     /# exit
90
91
92 10. docker-compose run
93     1)해당 서비스에 컨테이너 하나 더 실행할 때
94     $ docker-compose run -d nginx
95     $ docker-compose ps <---하나 더 컨테이너 있는 것을 확인
96     $ docker-compose run nginx bash
97     /# ls <--- TEST 파일 없음을 확인
98
99
100 11. docker-compose down
101     1)중지하고 삭제
102     $ docker-compose down
103
```