

```
1 1. docker-compose pull
2 -pull 뒤에 이미지를 적지 않으면 모든 이미지를 다운로드한다.
3 1)$ cat docker-compose.yml
4
5     version: '3'
6     services:
7       nginx:
8         image: nginx:latest
9         ports:
10          - 8000:80
11
12 2)$ docker-compose pull
13
14
15 2. docker-compose build
16 -build 뒤에 이미지를 적지 않으면 필요한 이미지를 모두 빌드한다.
17 1)index.html
18     <font size='7' color='blue'>Hello, Docker Compose</font>
19
20 2)Dockerfile
21     FROM      nginx:latest
22
23     RUN        ["chmod", "755", "/usr/share/nginx/html"]
24     RUN        ["chmod", "644", "/usr/share/nginx/html/index.html"]
25     ADD        ./index.html /usr/share/nginx/html/
26
27
28 3)$ cat docker-compose.yml
29     version: '3'
30     services:
31       nginx:
32         build:
33           context: . <---context가 현재 위치를 의미하고 있음. 같은 위치에 있는 Dockerfile을 사용할 것임.
34           dockerfile: ./dockerfile
35         ports:
36           - 8000:80
37
38 4)$ docker-compose build
39
40
41 3. docker-compose up
42 -pull과 build를 실행했기 때문에 서비스를 구동한다.
43 1)$ docker-compose up
44     -브라우저에서 확인
45     --Ctrl + C로 중지
46
47 2)$ docker-compose up --build
48     -필요한 이미지를 강제로 빌드
49
50 3)$ docker-compose up --force-recreate
51     -이미 실행중인 컨테이너에 뭔가 문제가 생겼거나 docker-compose file을 수정했다면 컨테이너를 새로 생성하면 된다.
52
53 4)$ docker-compose up -d
54     -서비스를 background로 실행, 실행 후 다시 터미널로 돌아옴.
55
56
57 4. docker-compose ps
58 1)서비스가 잘 실행되고 있는지 확인
59 2)Name의 앞부분은 현재 실행되고 있는 디렉토리
60     $ docker-compose ps
61
62
63 5. docker-compose logs
64 1)컨테이너별 로그를 보고 싶을 때
65     -$ docker-compose logs
66
67     -$ docker-compose logs -f
68     --브라우저에서 Refresh로 log가 계속 update되는 것을 확인
69
70 6. docker-compose top
71     $ docker-compose top
72     -지금 실행하고 있는 nginx의 Process를 확인
73
74
75 7. docker-compose stop
76     $ docker-compose stop
77     -stop뒤에 서비스 이름을 넣지 않으면 모든 서비스 중단
78
79     $ docker-compose ps
80     -중단된 서비스 확인
81     -State가 Exit됨을 확인
82
83
84 8. docker-compose start
```

```

85 -멈춘 서비스 다시 실행
86 $ docker-compose start
87
88
89 9. docker-compose exec
90 1)서비스 안에서 어떤 명령을 실행
91 -$ docker-compose exec nginx bash
92 /# touch TEST <---run과의 명령어 차이를 구별하기 위해 빈 파일 하나 생성
93 /# ls <---TEST 파일 확인
94 /# exit
95
96
97 10. docker-compose run
98 1)해당 서비스에 컨테이너 하나 더 실행할 때
99 $ docker-compose run -d nginx
100 $ docker-compose ps <---하나 더 컨테이너 있는 것을 확인
101 $ docker-compose run nginx bash
102 /# ls <--- TEST 파일 없음을 확인
103
104
105 11. docker-compose down
106 1)중지하고 삭제
107 $ docker-compose down
108
109 2)Error 발생시
110 $ docker compose down
111 [+] Running 1/0
112 x Network ubuntu_default Error 0.0s
113 failed to remove network ubuntu_default: Error response from daemon: error while removing network: network
ubuntu_default id a17bac259bb0bd387f51f3dea20382bdcf25ae086a1e366bc88af2bbebb0a5e6 has active endpoints
114
115 -다음의 명령을 사용
116 $ docker-compose down --remove-orphans
117 [+] Running 3/2
118 ✓ Container ubuntu-nginx-run-ad778dfe067a Removed 0.2s
119 ✓ Container ubuntu-nginx-run-95e8d6846106 Removed 0.0s
120 ✓ Network ubuntu_default Removed 0.1s

```