

```

1 1. Ubuntu기반 git 설치의 이미지 생성하기
2   1)Dockerfile
3     FROM ubuntu:latest
4
5     RUN apt-get update
6     RUN apt-get install -y git
7
8   2)Image Build
9     $ docker build -t ubuntu:git-dockerfile .
10    $ docker images
11
12   3)Container 생성하기
13     $ docker run -it --name git3 ubuntu:git-dockerfile bash
14     /# git --version
15
16
17 2. 실습
18   1)Dockerfile 작성하기
19     $ mkdir sample
20     $ cd sample
21     $ vim dockerfile
22       FROM centos:7
23       COPY name.dat .
24       CMD cat ./name.dat
25
26     $ cat > name.dat
27     Hello, Instructor!!!
28     Ctrl + Z
29     $ cat name.dat
30
31   2)Dockerfile 빌드하기
32     $ sudo docker build -t {{dockerhub 계정}}/dockerfiledemo:v1 .
33     Successfully built d7f2b162a692
34     Successfully tagged pythonexpert/dockerfiledemo:v1
35
36     $ sudo docker images
37
38   3)Container 실행하기
39     $ sudo docker run {{dockerhub 계정}}/dockerfiledemo:v1
40     Hello, Instructor!!!
41
42     $ sudo docker ps -a
43
44   4)Dockerfile 수정
45     $ vim dockerfile
46       FROM centos:7
47       COPY name.dat .
48       CMD while true; do sleep 3; cat ./name.dat; done;
49
50     $ sudo docker build -t {{dockerhub 계정}}/dockerfiledemo:v2 .
51     Successfully built 14c944b5ec08
52     Successfully tagged pythonexpert/dockerfiledemo:v2
53
54     -3초마다 Hello, Instructor!!! 출력
55
56   5)또 다른 세션에서
57     $ sudo docker ps -a
58     $ sudo docker exec -it pid bash
59     /# ls
60     /# cat name.dat

```

```

61     /# vim name.dat
62     Hello, World
63
64     /#exit
65
66 6)원 세션에서도 변경된 텍스트 출력확인
67     Hello, World
68     $ sudo docker stop pid
69
70
71 3. 실습
72 1)Dockerfile 생성하기
73     $ mkdir hellojs
74     $ cd hellojs
75     $ cat hello.js
76
77     var http = require('http');
78
79     var server = http.createServer();
80
81     server.addListener('request', function(request, response) {
82         console.log('requested...');
83         response.writeHead(200, {'Content-Type' : 'text/plain'});
84         response.writeLine('Hello, nodejs!!!');
85         response.end();
86     });
87
88     server.addListener('connection', function(socket){
89         console.log('connected...');
90     });
91
92     server.listen(8888);
93
94 $ vi dockerfile
95     FROM node:12
96     COPY hello.js /
97     CMD ["node", "/hello.js"]
98
99 $ sudo docker build -t hellojs:latest .
100
101 $ sudo docker images
102
103 $ docker run -d -p 8080:8888 --name web hellojs
104 $ curl localhost:8080
105
106 2)Ubuntu 기반의 Web Server Container 만들기
107     -DockerHub에서 'httpd'로 검색
108
109     $ mkdir webserver
110     $ cd webserver
111     $ vim dockerfile
112
113     FROM ubuntu:18.04
114     LABEL maintainer="instructor <javaexpert@nate.com>"
115
116     # Install Apache2
117     RUN apt update \
118         && apt install -y apache2
119     RUN echo "<body><h1>Hello Apache2</h1></body>" >
        /var/www/html/index.html

```

```
120
121     EXPOSE 80
122     CMD ["/usr/sbin/apachectl", "-DFOREGROUND"]
123
124     $ sudo docker build -t webserver:v1 .
125     $ sudo docker image ls
126
127     $ sudo docker run -d -p 80:80 --name web webserver:v1
128     $ curl localhost:80
129
130     $ sudo docker rm -f web
131     $ sudo docker ps -a
132     $ sudo docker images
133
134
135 3)Container Image 배포하기
136     $ sudo docker login
137     Username :
138     Password :
139
140     Login Succeeded
141     $ sudo docker images
142
143     $ sudo docker tag webserver:v1 {{dockerhub 계정}}/webserver:v1
144     $ sudo docker images
145
146     $ sudo docker push {{dockerhub 계정}}/webserver:v1
147
148     DockerHub/{{dockerhub 계정/repositories에서 확인할 것
149
150     $ cd ..
151     $ cd hellojs
152
153     $ sudo docker tag hellojs {{dockerhub 계정}}/hellojs
154     $ sudo docker images
155
156     $ sudo docker push {{dockerhub 계정}}/hellojs
157
158     DockerHub/{{dockerhub 계정/repositories에서 확인할 것
159
```