

```

1 1. Ubuntu기반 git 설치의 이미지 생성하기
2  $ mkdir demo
3  $ cd demo
4  $ docker system prune -a    <---Simply run to remove any stopped containers.
5
6 1)Dockerfile 생성
7  $ vim Dockerfile
8
9      FROM ubuntu:latest
10
11      RUN apt-get update
12      RUN apt-get install -y git
13
14 2)Image Build
15  $ docker build -t ubuntu:git-dockerfile .
16  $ docker images
17
18 3)Container 생성하기
19  $ docker run -it --name git3 ubuntu:git-dockerfile bash
20  /# git --version
21  git version 2.34.1
22
23
24 2. Lab
25 1)Dockerfile 작성하기
26  $ mkdir sample
27  $ cd sample
28  $ vim dockerfile
29      FROM centos:7
30      COPY name.dat .
31      CMD cat ./name.dat
32
33  $ cat > name.dat
34  Hello, World
35  Ctrl + Z
36  $ cat name.dat
37
38
39 2)Dockerfile 빌드하기
40  $ docker build -t {{dockerhub 계정}}/dockerfiledemo:v1 .
41  $ docker images
42
43
44 3)Container 실행하기
45  $ docker run {{dockerhub 계정}}/dockerfiledemo:v1
46  Hello, World
47
48  $ docker ps -a
49
50
51 4)Dockerfile 수정
52  $ vim dockerfile
53      FROM centos:7
54      COPY name.dat .
55      CMD while true; do sleep 3; cat ./name.dat; done;
56
57  $ docker build -t {{dockerhub 계정}}/dockerfiledemo:v2 .
58  $ docker run {{dockerhub 계정}}/dockerfiledemo:v1
59  -3초마다 Hello, World 출력
60
61 5)또 다른 세션에서
62  $ docker ps -a
63  $ docker exec -it {{ContainerID}} bash
64  /# ls
65  /# cat name.dat
66  /# vim name.dat
67  Hello, Docker World!!!
68
69  /#exit
70
71 6)원래의 세션에서도 변경된 텍스트 출력확인
72  Hello, Docker World!!!
73  $ docker stop {{ContainerID}}
74
75
76 3. Lab
77 1)Dockerfile 생성하기
78  $ mkdir hellojs
79  $ cd hellojs
80  $ cat hello.js
81
82      const http = require('http');
83
84      const server = http.createServer();

```

```

85
86     server.addListener('request', function(request, response) {
87         console.log('requested...');
88         response.writeHead(200, {'Content-Type' : 'text/plain'});
89         response.write('Hello, nodejs!!!');
90         response.end();
91     });
92
93     server.addListener('connection', function(socket){
94         console.log('connected...');
95     });
96
97     server.listen(8888);
98
99 $ vi dockerfile
100     FROM node:18    <---Docker Hub에서 검색해서 버전확인
101     COPY hello.js /
102     CMD ["node", "/hello.js"]
103
104 $ docker build -t hellojs:latest .
105
106 $ docker images
107
108 $ docker run -d -p 8080:8888 --name web hellojs
109 $ curl localhost:8080
110
111
112 2)Ubuntu 기반의 Web Server Container 만들기
113 -DockerHub에서 'httpd'로 검색
114
115     $ mkdir webserver
116     $ cd webserver
117     $ nano dockerfile
118
119     FROM ubuntu:latest
120     LABEL maintainer="instructor <javaexpert@nate.com>"
121
122     # Install Apache2
123     RUN apt update \
124         && apt install -y apache2
125     RUN echo "<body><h1>Hello Apache2</h1></body>" > /var/www/html/index.html
126
127     EXPOSE 80
128     CMD ["/usr/sbin/apache2ctl", "-DFOREGROUND"]
129
130 $ docker build -t webserver:v1 .
131 $ docker image ls
132
133 $ docker run -d -p 80:80 --name web webserver:v1
134 $ curl localhost:80
135
136 $ docker rm -f web
137 $ docker ps -a
138 $ docker images
139
140
141 3)Container Image 배포하기
142 $ docker login
143 Username :
144 Password :
145
146 Login Succeeded
147 $ docker images
148
149 $ docker tag webserver:v1 {{dockerhub 계정}}/webserver:v1
150 $ docker images
151
152 $ docker push {{dockerhub 계정}}/webserver:v1
153
154 DockerHub/{{dockerhub 계정}/repositories에서 확인할 것
155
156 $ cd ..
157 $ cd hellojs
158
159 $ docker tag hellojs {{dockerhub 계정}}/hellojs
160 $ docker images
161
162 $ docker push {{dockerhub 계정}}/hellojs
163
164 DockerHub/{{dockerhub 계정}/repositories에서 확인할 것

```