

```

1 1. Ubuntu기반 git 설치의 이미지 생성하기
2   1)Dockerfile 생성
3     FROM ubuntu:latest
4
5     RUN apt-get update
6     RUN apt-get install -y git
7
8   2)Image Build
9     $ sudo docker build -t ubuntu:git-dockerfile .
10    $ sudo docker images
11
12   3)Container 생성하기
13     $ sudo docker run -it --name git3 ubuntu:git-dockerfile bash
14     /# git --version
15
16
17 2. Lab
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. Lab
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 <---Docker Hub에서 검색해서 버전확인
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
107 2)Ubuntu 기반의 Web Server Container 만들기
108 -DockerHub에서 'httpd'로 검색
109
110 $ mkdir webserver
111 $ cd webserver
112 $ nano dockerfile
113
114     FROM ubuntu:18.04
115     LABEL maintainer="instructor <javaexpert@nate.com>"
116
117     # Install Apache2
118     RUN apt update \
119         && apt install -y apache2
120     RUN echo "<body><h1>Hello Apache2</h1></body>" > /var/www/html/index.html
121
122     EXPOSE 80
123     CMD ["/usr/sbin/apache2ctl", "-DFOREGROUND"]
124
125 $ sudo docker build -t webserver:v1 .
126 $ sudo docker image ls
127
128 $ sudo docker run -d -p 80:80 --name web webserver:v1
129 $ curl localhost:80
130
131 $ sudo docker rm -f web
132 $ sudo docker ps -a
133 $ sudo docker images
134

```

3) Container Image 배포하기

```
$ sudo docker login
```

```
Username :
```

```
Password :
```

```
Login Succeeded
```

```
$ sudo docker images
```

```
$ sudo docker tag webserver:v1 {{dockerhub 계정}}/webserver:v1
```

```
$ sudo docker images
```

```
$ sudo docker push {{dockerhub 계정}}/webserver:v1
```

DockerHub/{{dockerhub 계정}/repositories에서 확인할 것

```
$ cd ..
```

```
$ cd hellojs
```

```
$ sudo docker tag hellojs {{dockerhub 계정}}/hellojs
```

```
$ sudo docker images
```

```
$ sudo docker push {{dockerhub 계정}}/hellojs
```

DockerHub/{{dockerhub 계정}/repositories에서 확인할 것

4. Lab

1) Container 이름 : fortune:20.02

2) dockerfile의 내용

-Base Image : debian

-Container에 아래의 webpage.sh 파일 복사할 것

```
#!/bin/bash
```

```
mkdir /htdocs
```

```
while :
```

```
do
```

```
    /usr/games/fortune > /htdocs/index.html
```

```
    sleep 10
```

```
done
```

-Container에 fortune application 설치

```
--apt-get install fortune
```

-Container 실행시 저장한 webpage.sh 실행하도록

3) 실습

```
$ mkdir fortune
```

```
$ cd fortune
```

```
$ nano webpage.sh
```

```
#!/bin/bash
```

```
mkdir /htdocs
```

```
while :
```

```
do
```

```
    /usr/games/fortune > /htdocs/index.html
```

```
    sleep 10
```

```
done
```

```
$ nano dockerfile
```

```
FROM debian:latest
```

```
COPY webpage.sh /
```

```
RUN apt-get update && apt-get install -y fortune
```

```
RUN ["chmod", "+x", "./webpage.sh"]
```

```
CMD ["./webpage.sh"]
```

```
$ sudo docker build -t fortune:21.02 .
```

```
202
203 $ sudo docker run -d ---name fortune fortune:21.02
204
205 $ sudo docker exec -it fortune /bin/bash
206 # cat /htdocs/index.html
```