1. Ubuntu기반 git 설치의 이미지 생성하기

1)Dockerfile

```
FROM ubuntu:latest

RUN apt-get update
RUN apt-get install -y git
```

2)Image Build

```
$ docker build -t ubuntu:git-dockerfile .
$ docker images
```

3)Container 생성하기

```
$ docker run -it --name git3 ubuntu:git-dockerfile bash
/# git --version
```

2. 실습

1)Dockerfile 작성하기

```
$ mkdir sample
$ cd sample
$ vim dockerfile
    FROM centos:7
    COPY name.dat .
    CMD  cat ./name.dat

$ cat > name.dat
Hello, Instructor!!!
Ctrl + Z
$ cat name.dat
```

2)Dockerfile 빌드하기

```
$ sudo docker build -t {{dockerhub 계정}}/dockerfiledemo:v1 .
Successfully built d7f2b162a692
Successfully tagged pythonexpert/dockerfiledemo:v1

$ sudo docker images
```

3)Container 실행하기

```
$ sudo docker run {{dockerhub 계정}}/dockerfiledemo:v1
Hello, Instructor!!!

$ sudo docker ps -a
```

4)Dockerfile 수정

```
$ vim dockerfile
    FROM centos:7
    COPY name.dat .
    CMD  while true;  do sleep 3; cat ./name.dat;  done;

$ sudo docker build -t {{dockerhub 계정}}/dockerfiledemo:v2 .
Successfully built 14c944b5ec08
Successfully tagged pythonexpert/dockerfiledemo:v2
```

-3초마다 Hello, Instructor!!! 출력

5)또 다른 세션에서

```
$ sudo docker ps -a
$ sudo docker exec -it pid bash
/# ls
/# cat name.dat
/# vim name.dat
Hello, World

/#exit
```

6)원 세션에서도 변경된 텍스트 출력확인

```
Hello, World
$ sudo docker stop pid
```

3. 실습

1)Dockerfile 생성하기

```
$ mkdir hellojs
$ cd hellojs
$ cat hello.js

    var http = require('http');

    var server = http.createServer();

    server.addListener('request', function(request, response) {
        console.log('requested...');
        response.writeHead(200, {'Content-Type' : 'text/plain'});
        response.writeLine('Hello, nodejs!!!');
        response.end();
    });

    server.addListener('connection', function(socket){
        console.log('connected...');
    });

    server.listen(8888);

$ vi dockerfile
    FROM node:12
    COPY hello.js /
    CMD ["node", "/hello.js"]

$ sudo docker build -t hellojs:latest  .

$ sudo docker images

$ docker run -d -p 8080:8888 --name web  hellojs
$ curl localhost:8080
```

2)Ubuntu 기반의 Web Server Container 만들기

-DockerHub에서 'httpd'로 검색

```
$ mkdir webserver
$ cd webserver
$ vim dockerfile

    FROM ubuntu:18.04
    LABEL maintainer="instructor <javaexpert@nate.com>"

    #  Install Apache2
    RUN apt update \
          && apt install -y apache2
    RUN echo "<body><h1>Hello Apache2</h1></body>" > /var/www/html/index.html
```

```
            EXPOSE 80
            CMD ["/usr/sbin/apachectl", "-DFOREGROUND"]

        $ sudo docker build -t webserver:v1 .
        $ sudo docker image ls

        $ sudo docker run -d -p 80:80 --name web webserver:v1
        $ curl localhost:80

        $ sudo docker rm -f web
        $ sudo docker ps -a
        $ sudo docker images
```

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