

# Dockerfile을 이용하여 Docker image build 실습

B411001 강도연

```
ubuntu@ip-172-31-46-150:~$ mkdir dockerfile
ubuntu@ip-172-31-46-150:~$ cd dockerfile
ubuntu@ip-172-31-46-150:~/dockerfile$ vim doeckerfile
ubuntu@ip-172-31-46-150:~/dockerfile$ ls -la
total 12
drwxrwxr-x  2 ubuntu ubuntu 4096 Apr 20 05:38 .
drwxr-xr-x 23 ubuntu ubuntu 4096 Apr 20 05:38 ..
-rw-rw-r--  1 ubuntu ubuntu  137 Apr 20 05:38 doeckerfile
ubuntu@ip-172-31-46-150:~/dockerfile$ test.html
test.html: command not found
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ ls -la
total 16
drwxrwxr-x  2 ubuntu ubuntu 4096 Apr 20 05:44 .
drwxr-xr-x 23 ubuntu ubuntu 4096 Apr 20 05:44 ..
-rw-rw-r--  1 ubuntu ubuntu  137 Apr 20 05:38 doeckerfile
-rw-rw-r--  1 ubuntu ubuntu  147 Apr 20 05:44 test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ vim doeckerfile
ubuntu@ip-172-31-46-150:~/dockerfile$ vim dockerfile
ubuntu@ip-172-31-46-150:~/dockerfile$ ls -la
total 20
drwxrwxr-x  2 ubuntu ubuntu 4096 Apr 20 05:45 .
drwxr-xr-x 23 ubuntu ubuntu 4096 Apr 20 05:45 ..
-rw-rw-r--  1 ubuntu ubuntu  138 Apr 20 05:45 dockerfile
-rw-rw-r--  1 ubuntu ubuntu  137 Apr 20 05:38 doeckerfile
-rw-rw-r--  1 ubuntu ubuntu  147 Apr 20 05:44 test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ rm doeckerfile
ubuntu@ip-172-31-46-150:~/dockerfile$ ls -la
```

```
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker build . -t image/example
Sending build context to Docker daemon  3.072kB
Step 1/6 : FROM ubuntu
--> 4e5021d210f6
Step 2/6 : RUN apt-get update
--> Running in d8d9e0c88e67
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [889
```

Docker build command를 통해 image를 build 하였으며, . 의 의미는 현재 directory의 dockerfile을 이용하여 만들겠다는 뜻이다.

Dockerfile 디렉토리를 새로 만들고, dockerfile과 html file을 우선 만들었다.

Docker image가 성공적으로 tag와 함께 build 가 되었고, 이 image를 docker run command를 이용하여 동작시켜 보았다.  
Port는 EC2로 5000을 통해 들어오면 80으로 주도록 설정하였다.

```
Successfully built 61b16f51ea7d
Successfully tagged image/example:latest
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker run -it -p 5000:80 -d image/example
f66f3fd405eae510c7a02c32a2b9531b6be96d05ceddc175478aeb4af9a051f3
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker example
docker: 'example' is not a docker command.
See 'docker --help'
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
```

결과는 오른쪽과 같이 나왔다.



현재 상태에서 **html file**을  
수정해보았지만, **page**에는  
적용되지 않았다.

## This is B411001 Doyeun's Web Server

Thank you for teaching us!

```
ubuntu@ip-172-31-46-150: ~/dockerfile
Up 7 days
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker stop ^C
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker stop c4c661a3807f
c4c661a3807f
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker stop ^C
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker stop 7a54624d9cb5
7a54624d9cb5
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
f66f3fd405ea       image/example      "/bin/sh -c 'apachec..."   3 minutes
ago                0.0.0.0:5000->80/tcp hopeful_lamport
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$
```

이미 만들어진 **image**로부터  
가져오기 때문에 수정을 해도  
적용이 안되는 결과를 가져와  
서, **container**를 지우기로 하  
였다.

```
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
f66f3fd405ea       image/example      "/bin/sh -c 'apachec..."   3 minutes
ago                0.0.0.0:5000->80/tcp hopeful_lamport
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker rm -f ^C
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker rm -f f66f3fd405ea
f66f3fd405ea
```

사이트에 연결할 수 없음

3.134.77.3에서 연결을 거부했습니다.

다음 방법을 시도해 보세요.

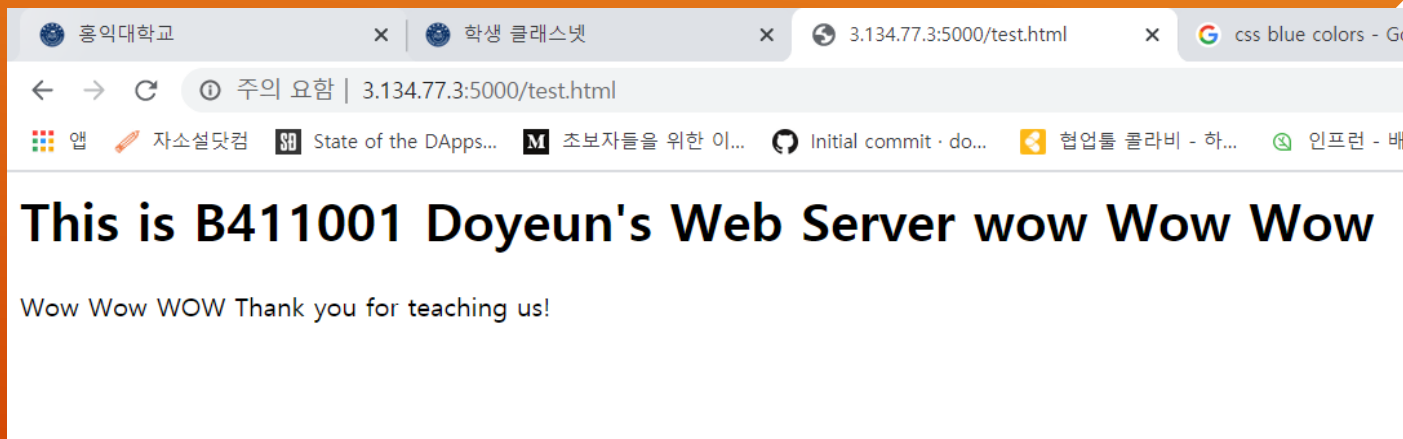
연결 확인

이번에 build 할때, source directroy를 destination director에 -v command를 이용하여 mount하였다. Mount할 destinatio과 source directory를 적어야하며, 둘 사이는 : 으로 나눈다.

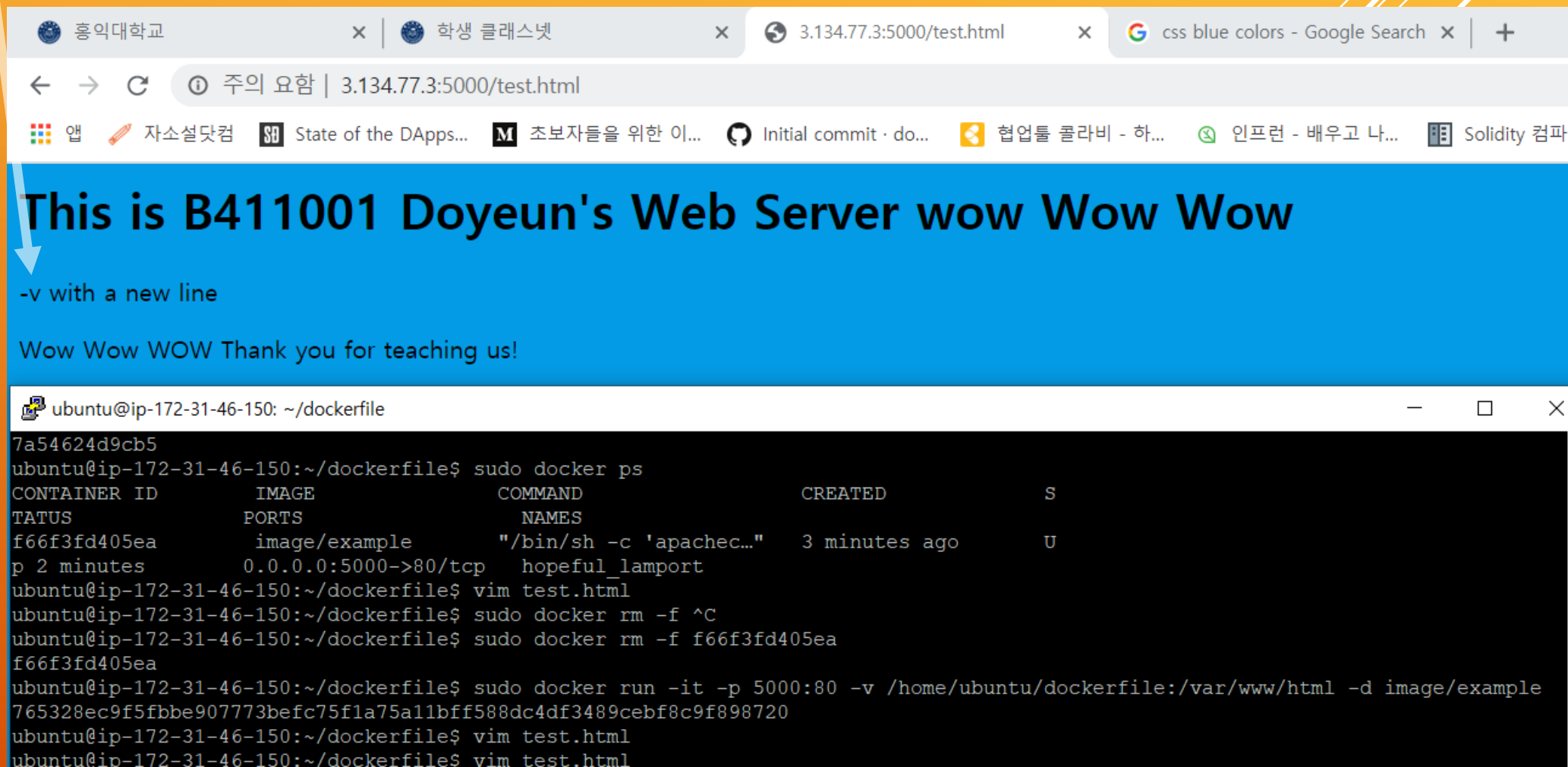
이렇게 되면, 내가 수정하는 즉시, 바로 적용이 된다.

Port 설정은 이전과 동일하다.

```
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker run -it -p 5000:80 -v /home/ubuntu/dockerfile:/var/www/html -d image/example
765328ec9f5fbbe907773befc75f1a75a11bff588dc4df3489cebf8c9f898720
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$
```



결과) html file을 수정하는 즉시, 그 변화가 적용된다.



The screenshot shows a web browser window with the URL `3.134.77.3:5000/test.html`. The browser displays a blue page with the text "This is B411001 Doyeun's Web Server wow Wow Wow". Below this, it says "-v with a new line" and "Wow Wow WOW Thank you for teaching us!". A blue arrow points from the text "결과) html file을 수정하는 즉시, 그 변화가 적용된다." to the first line of the terminal output.

```
ubuntu@ip-172-31-46-150: ~/dockerfile
7a54624d9cb5
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS
f66f3fd405ea        image/example       "/bin/sh -c 'apachec..." 3 minutes ago      Up                  0.0.0.0:5000->80/tcp
p 2 minutes
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker rm -f ^C
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker rm -f f66f3fd405ea
f66f3fd405ea
ubuntu@ip-172-31-46-150:~/dockerfile$ sudo docker run -it -p 5000:80 -v /home/ubuntu/dockerfile:/var/www/html -d image/example
765328ec9f5fbbe907773befc75f1a75a11bff588dc4df3489ceb8c9f898720
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
ubuntu@ip-172-31-46-150:~/dockerfile$ vim test.html
```