

```

1 [HOL] Docker Container
2
3 1. Docker Hub에서 Container Image 검색하기
4 1)Docker Version 확인
5 $ sudo docker version
6
7 2)Docker Service 확인하기
8 $ su -
9 # systemctl status docker
10
11 3)Docker Hub에서 nginx 검색하기
12 # docker search nginx
13
14
15 2. Container Image 다운로드 후 Image Layer 보기
16 1)Docker Layer 확인하기
17 # cd /var/lib/docker
18 # ls -l
19 # cd overlay2
20 # ls -l
21 # cd /home/{계정}
22 # docker images
23
24 2)Docker Hub에서 Nginx Pull
25 # docker pull nginx:latest
26 # docker images
27 # ls -l /var/lib/docker/overlay2/ <---6개의 directory 확인
28
29
30 3. Container 실행하고 확인하기
31 1)Docker Image 확인
32 # docker image ls
33
34 2)Docker Image 실행하기
35 # docker run -d --name webserver -p 80:80 nginx:latest
36 # curl localhost:80
37
38 3)docker Container Stop
39 # docker ps
40 # docker stop webserver
41 # docker ps -a
42
43 4)docker Container remove
44 # docker rm webserver
45 # docker ps -a
46
47 5)docker Image remove
48 # docker image ls
49 # docker rmi nginx
50 # docker images
51 # ls -l /var/lib/docker/overlay2/
52
53
54 4. Port Binding 하기
55 1)Server 단에서 Nginx 실행하기
56 # docker run -p 80:80 nginx
57 log 대기
58
59 2)Client 단에서
60 $ curl localhost:80
61
62 -Server 단의 logging
63 172.17.0.1 - - [21/Jun/2021:06:02:26 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.68.0" "-"
64
65 3)Client 단에서 404 Not Found 페이지 호출
66 $ curl localhost:80/aaa.html
67

```

```

68 -Server 단에서 에러 Logging
69 172.17.0.1 - - [21/Jun/2021:06:04:52 +0000] "GET /aaa.html HTTP/1.1" 404 153 "-"
    "curl/7.68.0" "-"
70 2021/06/21 06:04:52 [error] 31#31: *3 open() "/usr/share/nginx/html/aaa.html" failed (2: No
    such file or directory), client: 172.17.0.1, server: localhost, request: "GET /aaa.html HTTP/1.1",
    host: "localhost"
71
72 Ctrl + C <---- Server단에서 Service 중지
73
74 -Client 단에서 호출
75 $ curl localhost:80/aaa.html
76 curl : (7) Failed to connect to localhost port 80: Connection refused
77
78 4)Port binding 하기
79 -Server단에서 nginx 실행
80   # docker run -p 8080:80 nginx
81   log 대기
82
83 -Client 단에서 접속
84   $ curl localhost:8080
85
86 -만일 $ curl localhost:80으로 연결하면
87 curl : (7) Failed to connect to localhost port 80: Connection refused
88
89
90 5. Docker Volume Mount하기
91 1)Server 단에서 MongoDB search
92   $ docker search mongodb
93
94 2)Server 단에서 MongoDB 실행하기
95   $ sudo docker run -v ${PWD}/data:/data/db mongo:4
96
97 3)Client 단에서 접속하기
98   $ ls -al
99   $ cd ./data
100  $ ls <----여러개의 파일과 디렉토리 확인
101  $ sudo docker ps <--MongoDB PID 확인
102
103  $ sudo docker exec -it PID(앞 2자리도 가능) mongo
104  >
105  > show dbs;
106  admin
107  config
108  local
109
110  >use example
111  switched to db example
112  >db.example.insert({"name":"Henry Instructor"})
113  WriteResult({"nInserted" : 1})
114
115  >db.example.find({})
116  ....
117  >exit
118
119  $ Server 단에서 Ctrl + C 로 서비스 정지
120
121 4)다시 Docker Run을 했을 때 Data가 남아 있을 것인가?
122 -Server단에서 MongoDB 실행
123   $ sudo docker run -v ${PWD}/data:/data/db mongo:4
124
125 -Client 단에서 접속
126   $ sudo docker ps <--- PID확인
127
128   $ sudo docker exec -it PID(앞 2자리도 가능) mongo
129   >show dbs
130   <--- example db 확인
131

```

```

132         >use example
133         >db.example.find({})
134         <-- 앞에서 저장한 데이터 확인
135
136 5)MongoDB Image 모두 삭제
137 6)다시 Server 단에서 MongoDB Image Run
138
139     $ sudo docker run mongo:4
140
141 7)Client 단에서 접속
142     $ sudo rm -rf ./data
143     $ sudo docker exec -it PID mongo
144     >show dbs
145     >use example
146     >db.example.insert({"name" : "Henry Instructor"})
147     >db.example.find({})
148     >exit
149
150     -MongoDB Server도 Ctrl + C로 서비스 정지
151
152 8)다시 MongoDB Server Start
153     $ sudo docker run mongo:4
154
155 9)Client 단에서 접속
156     $ sudo docker exec -it PID mongo
157     >
158     >show dbs
159     <---example db 없음.
160
161
162 6. Container Image 삭제하기
163 1)Server 단에서 redis 실행하기
164     # docker run -p 6379:6379 redis
165
166 2)클라이언트 단에서
167     $ sudo apt install redis-tools
168     $ redis-cli
169     127.0.0.1:6379>set name "Henry Instructor"
170     OK
171     127.0.0.1:6379>get name
172     "Henry Instructor"
173     127.0.0.1:6379>exit
174
175     $ sudo docker ps -a <-- PID 확인
176     $ sudo docker rm PID --> 실패, 이유는 현재 Docker Container 실행 중
177     $ sudo docker stop PID <---클라이언트 세션에서 서버 서비스 중지시킴.
178
179 3)Container 삭제하기
180     $ sudo docker ps -a <--- PID확인
181     $ sudo docker rm PID
182
183 4)Container Image 삭제하기
184     # docker images <--- PID 확인
185     # docker rmi PID
186
187
188 7. MySQL 사용하기
189 1)Docker로 MySQL Run
190     $ mkdir mysql
191     $ cd mysql
192     $ su -
193     # docker pull mysql:5.7.34
194     # docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=password -d -p
195     3306:3306 mysql:5.7.34
196     # docker ps -a
197
198 2)MySQL Workbench 설치하기

```

-<https://dev.mysql.com/downloads/workbench/>  
-Windows (x86, 64-bit), MSI Installer 다운로드 후 설치

### 3)MySQL Workbench에서 Docker의 MySQL 연결하기

-MySQL Connection 추가  
--Connection Name : docker-mysql  
--Hostname : 192.168.56.101  
--Port : 3306  
--Username : root  
--Password : Store in Vault ... 클릭 > Password : password > OK  
--Test Connection Click  
--OK  
-docker-mysql double-click

### 4)Terminal 에서 연결하기

```
# docker exec -it mysql-container bash
# mysql -u root -p
Enter password : password
mysql > show databases;

mysql>exit
# exit
# docker rm -f mysql-container
```

## 8. Web Server를 만들어보기

### 1)Docker Image Pull

```
$ sudo docker pull httpd

$ sudo docker images
```

### 2)Docker Container 구동하기

-docker run 명령을 통해 Container 를 시작하고 Web 서비스를 구성 할 수 있다  
\$ sudo docker run httpd

-하지만, Container 가 Foreground 로 작동하면서 Shell 을 사용을 못할 뿐더러, Shell이 종료가 되면 httpd Container도 중지된다.

-위와 같이 되면, 전혀 서비스에 적용 할 수가 없다.

-그리하여 아래와 같이 background 로 container 를 실행하면 된다.

```
$ sudo docker run -d httpd
$ sudo docker ps -a
```

-Shell 에서 다른 명령도 가능하고 서비스가 계속 실행되는 것을 확인 할 수 있다.

-그럼 실제로 서비스가 작동하는지 확인해 본다.

```
$ curl http://127.0.0.1
curl: (7) Failed connect to 127.0.0.1; 연결이 거부됨.
```

-기존에 실행중이던 Container 중지

```
$ sudo docker stop [container ID]
$ sudo docker ps -a
```

-Port Binding

```
$ sudo docker run -d -p 80:80 httpd
$ sudo docker ps -a
```

-Service 확인하기

```
$ curl http://127.0.0.1
<html><body><h1>It works!</h1></body></html>
```

-Web Browser에서 확인할 것

### 3)index.html 수정하기

-Container 내부로 들어가서 index.html 수정하기

```
$ sudo docker -it [container ID] bash
root@419c02446fed:/# cd /usr/local/apache2/htdocs
root@419c02446fed:/usr/local/apache2/htdocs# cat index.html
<html><body><h1>It works!</h1></body></html>
root@419c02446fed:/usr/local/apache2/htdocs# echo "<html><body><h1>Docker Test
Page</h1></body></html>" > index.html
root@419c02446fed:/usr/local/apache2/htdocs# exit
exit
$ curl http://127.0.0.1
<html><body><h1>Docker Test Page</h1></body></html>
```

-Web Browser에서 확인할 것