

```

1 1. Docker Hub에서 Container Image 검색하기
2 2. Container Image 다운로드 후 Image Layer 보기
3 3. Container 실행하고 확인하기
4
5
6 1. Docker Hub에서 Container Image 검색하기
7 1)Docker Version 확인
8 $ docker version
9
10 2)Docker Service 확인하기
11 $ su -
12 # systemctl status docker
13
14 3)Docker Hub에서 nginx 검색하기
15 # docker search nginx
16
17
18 2. Container Image 다운로드 후 Image Layer 보기
19 1)Docker Layer 확인하기
20 # cd /var/lib/docker
21 # ls -l
22 # cd overlay2
23 # ls -l
24 # cd /home/{계정}
25 # docker images
26
27 2)Docker Hub에서 Nginx Pull
28 # docker pull nginx:latest
29 # docker images
30 # ls -l /var/lib/docker/overlay2/
31
32
33 3. Container 실행하고 확인하기
34 1)Docker Image 확인
35 # docker image ls
36
37 2)Docker Image 실행하기
38 # docker run -d --name webserver -p 80:80 nginx:latest
39 # curl localhost:80
40
41 3)docker Container Stop
42 # docker ps
43 # docker stop webserver
44 # docker ps -a
45
46 4)docker Container remove
47 # docker rm webserver
48 # docker ps -a
49
50 5)docker Image remove
51 # docker image ls
52 # docker rmi nginx
53 # docker images
54 # ls -l /var/lib/docker/overlay2/
55
56
57 4. Port Binding 하기
58 1)Server 단에서 Nginx 실행하기
59 # docker run -p 80:80 nginx
60 log 대기
61
62 2)Client 단에서
63 $ curl localhost:80
64
65 -Server 단의 logging
66 172.17.0.1 - - [21/Jun/2021:06:02:26 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.68.0" "-"
67

```

```

68 3)Client 단에서 404 Not Found 페이지 호출
69 $ curl localhost:80/aaa.html
70
71 -Server 단에서 에러 Logging
72 172.17.0.1 - - [21/Jun/2021:06:04:52 +0000] "GET /aaa.html HTTP/1.1" 404 153 "-"
    "curl/7.68.0" "-"
73 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"
74
75 Ctrl + C <---- Server단에서 Service 중지
76
77 -Client 단에서 호출
78 $ curl localhost:80/aaa.html
79 curl : (7) Failed to connect to localhost port 80: Connection refused
80
81 4)Port binding 하기
82 -Server단에서 nginx 실행
83   # docker run -p 8080:80 nginx
84   log 대기
85
86 -Client 단에서 접속
87 $ curl localhost:8080
88
89 -만일 $ curl localhost:80으로 연결하면
90 curl : (7) Failed to connect to localhost port 80: Connection refused
91
92
93 5. Docker Volume Mount하기
94 1)Server 단에서 MongoDB search
95 $ docker search mongodb
96
97 2)Server 단에서 MongoDB 실행하기
98 $ sudo docker run -v ${PWD}/data:/data/db mongo:4
99
100 3)Client 단에서 접속하기
101 $ ls -al
102 $ cd ./data
103 $ ls <----여러개의 파일과 디렉토리 확인
104 $ sudo docker ps <--MongoDB PID 확인
105
106 $ sudo docker exec -it PID(앞 2자리도 가능) mongo
107 >
108 > show dbs;
109 admin
110 config
111 local
112
113 >use example
114 switched to db example
115 >db.example.insert({"name":"Henry Instructor"})
116 WriteResult({"nInserted" : 1})
117
118 >db.example.find({})
119 ....
120 >exit
121
122 $ Server 단에서 Ctrl + C 로 서비스 정지
123
124 4)다시 Docker Run을 했을 때 Data가 남아 있을 것인가?
125 -Server단에서 MongoDB 실행
126 $ sudo docker run -v ${PWD}/data:/data/db mongo:4
127
128 -Client 단에서 접속
129 $ sudo docker ps <--- PID확인
130
131 $ sudo docker exec -it PID(앞 2자리도 가능) mongo

```

```

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

```

```
198 # docker ps -a
199
200 2)MySQL Workbench 설치하기
201 -https://dev.mysql.com/downloads/workbench/
202 -Windows (x86, 64-bit), MSI Installer 다운로드 후 설치
203
204 3)MySQL Workbench에서 Docker의 MySQL 연결하기
205 -MySQL Connection 추가
206 --Connection Name : docker-mysql
207 --Hostname : 192.168.56.101
208 --Port : 3306
209 --Username : root
210 --Password : Store in Vault ... 클릭 > Password : password > OK
211 --Test Connection Click
212 --OK
213 -docker-mysql double-click
214
215 4)Terminal 에서 연결하기
216 # docker exec -it mysql-container bash
217 # mysql -u root -p
218 Enter password : password
219 mysql > show databases;
220
221 mysql>exit
222 # exit
223 # docker rm -f mysql-container
```