[HOL] Docker Container

1. Docker Hub에서 Container Image 검색하기

1)Docker Version 확인

```
$ sudo docker version
Client: Docker Engine - Community
 Version:           23.0.4
 API version:       1.42
 Go version:        go1.19.8
 Git commit:        f480fb1
 Built:             Fri Apr 14 10:32:03 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          23.0.4
  API version:      1.42 (minimum version 1.12)
  Go version:       go1.19.8
  Git commit:       cbce331
  Built:            Fri Apr 14 10:32:03 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.20
  GitCommit:        2806fc1057397dbaeefbea0e4e17bddfbd388f38
 runc:
  Version:          1.1.5
  GitCommit:        v1.1.5-0-gf19387a
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

2)Docker Service 확인하기

```
$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-04-20 05:42:02 UTC; 1h 15min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 9466 (dockerd)
      Tasks: 9
     Memory: 69.2M
        CPU: 17.609s
     CGroup: /system.slice/docker.service
             └─9466 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 20 05:42:02 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T05:42:02.053352860Z" level=info msg="Daemon has completed initializ>Apr 20 05:42:02 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T05:42:02.079610211Z" level=info msg="[core] [Server #7] Server crea>Apr 20 05:42:02 ip-10-0-10-23 systemd[1]: Started Docker Application Container Engine.
Apr 20 05:42:02 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T05:42:02.089812011Z" level=info msg="API listen on /run/docker.sock"Apr 20 05:44:43 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T05:44:43.843267188Z" level=info msg="ignoring event" container=e22a>Apr 20 05:58:04 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T05:58:04.646802127Z" level=info msg="ignoring event" container=7d48>Apr 20 06:00:04 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T06:00:04.427063923Z" level=info msg="ignoring event" container=7d48>Apr 20 06:01:51 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T06:01:51.054431610Z" level=info msg="ignoring event" container=7d48>Apr 20 06:15:36 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T06:15:36.064292639Z" level=info msg="ignoring event" container=9fb2>Apr 20 06:27:46 ip-10-0-10-23 dockerd[9466]: time="2023-04-20T06:27:46.862748738Z" level=info msg="ignoring event" container=b51c>lines 1-22/22 (END)
```

3)Docker Hub에서 nginx 검색하기

```
$ docker search nginx
NAME                                              DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
nginx                                             Official build of Nginx.                        18407     [OK]
unit                                              Official build of NGINX Unit: a polyglot app…   0         [OK]
bitnami/nginx                                     Bitnami nginx Docker Image                      158                  [OK]
bitnami/nginx-ingress-controller                  Bitnami Docker Image for NGINX Ingress Contr…   25                   [OK]
ubuntu/nginx                                      Nginx, a high-performance reverse proxy & we…   84
kasmweb/nginx                                     An Nginx image based off nginx:alpine and in…   4
rancher/nginx-ingress-controller                                                                  11
rancher/nginx-ingress-controller-defaultbackend                                                   2
bitnami/nginx-exporter                                                                            3
rancher/nginx                                                                                     2
rapidfort/nginx-ib                                RapidFort optimized, hardened image for NGIN…   0
rapidfort/nginx                                   RapidFort optimized, hardened image for NGINX   3
vmware/nginx-photon                                                                               1
bitnami/nginx-ldap-auth-daemon                                                                    3
rapidfort/nginx-official                          RapidFort optimized, hardened image for NGIN…   1
vmware/nginx                                                                                      2
rancher/nginx-conf                                                                                0
```

```
    linuxserver/nginx                             An Nginx container, brought to you by LinuxS…   192
    privatebin/nginx-fpm-alpine                        PrivateBin running on an Nginx, php-fpm & Al…   72                   [OK]
    bitnami/nginx-intel                                                                       1
    rancher/nginx-ssl                                                                         0
    circleci/nginx                             This image is for internal use                 2
    continuumio/nginx-ingress-ws                                                                     0
    rancher/nginx-ingress-controller-amd64                                                              0
    webdevops/nginx                               Nginx container                                11                   [OK]
```

2. Container Image 다운로드 후 Image Layer 보기

1)/var/lib/docker/overlay2로 이동하여 'l' 디렉토리를 제외한 모든 디렉토리 삭제

2)# ls -l

```
drwxr-xr-x  2 root root 4096 Apr 20 07:12 l
```

3)# docker images

```
REPOSITORY    TAG       IMAGE ID       CREATED      SIZE
```

4)Docker Hub에서 Nginx Pull

```
    # docker pull nginx
    Using default tag: latest
    latest: Pulling from library/nginx
    26c5c85e47da: Pull complete
    4f3256bdf66b: Pull complete
    2019c71d5655: Pull complete
    8c767bdbc9ae: Pull complete
    78e14bb05fd3: Pull complete
    75576236abf5: Pull complete
    Digest: sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
    Status: Downloaded newer image for nginx:latest
    docker.io/library/nginx:latest
```

5)overlay2 디렉토리 이미지 확인

```
    $ sudo -i
    # cd /var/lib/docker
    # ls -l
    total 44
    drwx--x--x  4 root root 4096 Apr 20 05:42 buildkit
    drwx--x---  5 root root 4096 Apr 20 06:14 containers
    -rw-------  1 root root   36 Apr 20 05:42 engine-id
    drwx------  3 root root 4096 Apr 20 05:42 image
    drwxr-x---  3 root root 4096 Apr 20 05:42 network
    drwx--x--- 18 root root 4096 Apr 20 06:59 overlay2
    drwx------  4 root root 4096 Apr 20 05:42 plugins
    drwx------  2 root root 4096 Apr 20 05:42 runtimes
    drwx------  2 root root 4096 Apr 20 05:42 swarm
    drwx------  2 root root 4096 Apr 20 06:59 tmp
    drwx-----x  2 root root 4096 Apr 20 05:42 volumes

    # cd overlay2      <---6개의 directory 확인
    # ls -l
    total 36
    drwx--x---  9 root root 4096 Apr 20 07:12 .
    drwx--x--- 12 root root 4096 Apr 20 05:42 ..
    drwx--x---  4 root root 4096 Apr 20 07:12 0ac5167c28017a36965c7a6966fd1796847b481a91e65625bfeb32699adddf4f
    drwx--x---  4 root root 4096 Apr 20 07:12 3e35e04a44573a8c602a00a8bb19de77ec3dbaf9e4275959e46160b1d82c4a04
    drwx--x---  3 root root 4096 Apr 20 07:12 400eeb8965b353756cbb1d701cddfe8971d3a2be3d1e0ff055e5817cc10c3edc
    drwx--x---  4 root root 4096 Apr 20 07:12 501c0ec5e945199f922397d49476e14f8d8872077cdc3fc16b17568f7e3e6775
    drwx--x---  4 root root 4096 Apr 20 07:12 c640e945a0504fa09a73e82d6aa3701e103f8fe8ddd7f9dd48d6355f7689ea27
    drwx--x---  4 root root 4096 Apr 20 07:12 f6868cdd24c04b0d5c5b1465b1bce5577ce26b88e30f39baac0c27ea40337521
    drwxr-xr-x  2 root root 4096 Apr 20 07:12 l


    # cd /home/{{계정}}
    # docker images
    REPOSITORY    TAG       IMAGE ID       CREATED      SIZE
    nginx         latest    6efc10a0510f   7 days ago   142MB
```

3. Container 실행하고 확인하기

1)Docker Image 확인

```
    # docker image ls
    root@ip-10-0-10-23:/home/ubuntu# docker image ls
    REPOSITORY    TAG       IMAGE ID       CREATED      SIZE
    nginx         latest    6efc10a0510f   7 days ago   142MB
```

2)Docker Image 실행하기

```
    # docker run -d --name webserver -p 80:80 nginx:latest
    # curl localhost:80
    <!DOCTYPE html>
    <html>
```

```
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

※만일 위의 실행을 Cloud에서 수행하면 해당 가상머신의 인스턴스 보안 그룹에서 80번 포트를 열어서 확인 가능.

3)docker Container Stop

```
# docker ps
# docker stop werserver
# docker ps -a
```

4)docker Container remove

```
# docker rm webserver
# docker ps -a
```

5)docker Image remove

```
# docker image ls
# docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
Deleted: sha256:6efc10a0510f143a90b69dc564a914574973223e88418d65c1f8809e08dc0a1f
Deleted: sha256:a489ce38666d5aff5d73930d115381b154a503b48c5534357d5183160f5b9bfa
Deleted: sha256:ce2a611250a8bb55b73959de5865489d566bf48e1342c74b46b107e9a224370e
Deleted: sha256:fa6c51798227fd451cc3ff628c46c1b62c0d1d08e43374be7000c78ad910c0c0
Deleted: sha256:c426467d564c8684df87356eb045dc50d8b5af9521e9775c1b6bb941135680de
Deleted: sha256:c6deada06ccd386201e5b95c38e19297cd3eeb5264d6413d9cc7441020816401
Deleted: sha256:ed7b0ef3bf5bbec74379c3ae3d5339e666a314223e863c70644f7522a7527461

# docker images
# ls -l /var/lib/docker/overlay2/
total 4
drwxr-xr-x 2 root root 4096 Apr 20 07:23 l
```

4. Port Binding 하기

1)Server-side에서 Nginx 실행하기

```
# docker run -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
26c5c85e47da: Pull complete
4f3256bdf66b: Pull complete
2019c71d5655: Pull complete
8c767bdbc9ae: Pull complete
78e14bb05fd3: Pull complete
75576236abf5: Pull complete
Digest: sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/04/20 07:25:48 [notice] 1#1: using the "epoll" event method
2023/04/20 07:25:48 [notice] 1#1: nginx/1.23.4
2023/04/20 07:25:48 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/04/20 07:25:48 [notice] 1#1: OS: Linux 5.15.0-1031-aws
2023/04/20 07:25:48 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/04/20 07:25:48 [notice] 1#1: start worker processes
2023/04/20 07:25:48 [notice] 1#1: start worker process 29
                                <--- log 대기중
```

2)Client-side에서

```
$ curl localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

-Server-side에서 logging   <---또 다른 세션으로 접속

```
172.17.0.1 - - [20/Apr/2023:07:26:43 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/7.81.0" "-"
```

3)Client-side에서 404 Not Found 페이지 호출

```
$ curl localhost:80/aaa.html
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.23.4</center>
</body>
</html>
```

-Server-side에서 에러 Logging

```
2023/04/20 07:27:29 [error] 29#29: *2 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"
172.17.0.1 - - [20/Apr/2023:07:27:29 +0000] "GET /aaa.html HTTP/1.1" 404 153 "-" "curl/7.81.0" "-"
```

Ctrl + C <---- Server-side에서 Service 중지

-Client-side에서 호출

```
$ curl localhost:80/aaa.html
curl: (7) Failed to connect to localhost port 80 after 0 ms: Connection refused
```

4)Port binding 하기

-Server-side에서 nginx 실행

```
    # docker run -p 8080:80 nginx
    /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
    /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
    /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
    10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
    10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
    /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
    /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
    /docker-entrypoint.sh: Configuration complete; ready for start up
    2023/04/20 07:29:22 [notice] 1#1: using the "epoll" event method
    2023/04/20 07:29:22 [notice] 1#1: nginx/1.23.4
    2023/04/20 07:29:22 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
    2023/04/20 07:29:22 [notice] 1#1: OS: Linux 5.15.0-1031-aws
    2023/04/20 07:29:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
    2023/04/20 07:29:22 [notice] 1#1: start worker processes
    2023/04/20 07:29:22 [notice] 1#1: start worker process 29
                              <----- log 대기
```

-Client-side에서 접속

```
    $ curl localhost:8080
    <!DOCTYPE html>
    <html>
    <head>
    <title>Welcome to nginx!</title>
    <style>
    html { color-scheme: light dark; }
    body { width: 35em; margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif; }
    </style>
    </head>
    <body>
    <h1>Welcome to nginx!</h1>
    <p>If you see this page, the nginx web server is successfully installed and
    working. Further configuration is required.</p>

    <p>For online documentation and support please refer to
    <a href="http://nginx.org/">nginx.org</a>.<br/>
    Commercial support is available at
    <a href="http://nginx.com/">nginx.com</a>.</p>

    <p><em>Thank you for using nginx.</em></p>
    </body>
    </html>
```

-만일 $ curl localhost:80으로 연결하면

```
curl: (7) Failed to connect to localhost port 80 after 0 ms: Connection refused
```

5)Tomcat 설치하기

-Tomcat Search

```
$ docker search tomcat
NAME                                  DESCRIPTION                                        STARS     OFFICIAL   AUTOMATED
tomcat                                Apache Tomcat is an open source implementati…   3525      [OK]
tomee                                 Apache TomEE is an all-Apache Java EE certif…   106       [OK]
bitnami/tomcat                        Bitnami Tomcat Docker Image                     48                   [OK]
bitnamicharts/tomcat                                                                  0
secoresearch/tomcat-varnish           Tomcat and Varnish 5.0                          0                    [OK]
vulhub/tomcat                                                                         0
wnprcehr/tomcat                                                                       0
hivdb/tomcat-with-nucamino                                                            0
sismics/tomcat                        Apache Tomcat Servlet Container                 1
eclipse/rdf4j-workbench               Dockerfile for Eclipse RDF4J Server and Work…   6
semoss/docker-tomcat                  Tomcat, Java, Maven, and Git on top of debian   0                    [OK]
eclipse/hadoop-dev                    Ubuntu 14.04, Maven 3.3.9, JDK8, Tomcat 8       0                    [OK]
dhis2/base-dev                        Images in this repository contains DHIS2 WAR…   0
eclipse/alpine_jdk8                   Based on Alpine 3.3. JDK 1.8, Maven 3.3.9, T…   1                    [OK]
misolims/miso-base                    MySQL 5.7 Database and Tomcat 8 Server neede…   0
dhis2/base                            Images in this repository contains DHIS2 WAR…   0
jelastic/tomcat                       An image of the Tomcat Java application serv…   4
cfje/tomcat-resource                  Tomcat Concourse Resource                       2
rightctrl/tomcat                      CentOS , Oracle Java, tomcat application ssl…   7                    [OK]
amd64/tomcat                          Apache Tomcat is an open source implementati…   6
arm64v8/tomcat                        Apache Tomcat is an open source implementati…   8
tomcat2111/papercut-mf                PaperCut MF Application Server                  0
tomcatengineering/pg_backup_rotated   Clone of martianrock/pg_backup_rotated but w…   0
tomcat0823/auto1                                                                      0
softwareplant/tomcat                  Tomcat images for jira-cloud testing            0                    [OK]
```

-Tomcat Download

```
$ docker pull consol/tomcat-8.0
Using default tag: latest
latest: Pulling from consol/tomcat-8.0
Image docker.io/consol/tomcat-8.0:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
e5ad7970bc69: Pull complete
a3ed95caeb02: Pull complete
09fad3d8cd0f: Pull complete
e4b877670a03: Pull complete
7fe52da169a9: Pull complete
dd8c3151a5a5: Pull complete
70eb33b1a032: Pull complete
878a118528b8: Pull complete
e2c1f6b6c22a: Pull complete
0577651f52ea: Pull complete
494378cb5629: Pull complete
ae04179859b1: Pull complete
4167c9503f09: Pull complete
Digest: sha256:8107d4c293dd34524e46dd6e62a0370273cf8b8807587acb954f5724e90b6e20
Status: Downloaded newer image for consol/tomcat-8.0:latest
docker.io/consol/tomcat-8.0:latest
```

-Tomcat Run

```
$ docker run -d -p 8080:8080 consol/tomcat-8.0
```

-Tomcat Container check

```
$ docker ps -a
```

-Web Browser에서 확인하기 <--- 미리 보안그룹에서 인바운드 포트 8080 설정할 것

http://container-ip:8080

-Tomcat Manager

```
        http://container-ip:8080/manager/html
        -User : admin
        -Password : admin
```

5. Docker Volume Mount하기

1)Server 단에서 MongoDB search

```
$ docker search mongodb
NAME                                          DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
mongo                                         MongoDB document databases provide high avai…   9603      [OK]
mongo-express                                 Web-based MongoDB admin interface, written w…   1302      [OK]
mongodb/mongodb-atlas-kubernetes-operator     The MongoDB Atlas Kubernetes Operator - Kube…   2
mongodb/mongodb-atlas-kubernetes-operator-prerelease   This is an internal-use-only build of the Mo…   0
mongodb/mongodb-community-server              The Official MongoDB Community Server           12
bitnami/mongodb                               Bitnami MongoDB Docker Image                    214                  [OK]
mongodb/mongodb-enterprise-server             The Official MongoDB Enterprise Advanced Ser…   1
bitnami/mongodb-exporter                                                                      9
percona/mongodb_exporter                      A Prometheus exporter for MongoDB including …   3
bitnami/mongodb-sharded                                                                       11
rapidfort/mongodb                             RapidFort optimized, hardened image for Mong…   15
rancher/mongodb-conf                                                                          2
rapidfort/mongodb-ib                          RapidFort optimized, hardened image for Mong…   0
rapidfort/mongodb-official                    RapidFort optimized, hardened image for Mong…   1
rapidfort/mongodb-perfomance-test                                                             0
bitnamicharts/mongodb                                                                         0
bitnamicharts/mongodb-sharded                                                                 0
rancher/mongodb-config                                                                        0
drud/mongodb                                  Mongodb                                         0                    [OK]
kope/mongodb                                                                                  0
circleci/mongo                                CircleCI images for MongoDB                     13                   [OK]
percona/percona-server-mongodb                Percona Server for MongoDB docker images        37
edgexfoundry/docker-edgex-mongo               ARCHIVED! MongoDB container for older versio…   5
sensebox/opensensemap-api-mongo               Development MongoDB image for openSenseMap A…   0                    [OK]
noenv/mongo-exporter                          Prometheus MongoDB Exporter Docker Image        1
```

2)Server 단에서 MongoDB 실행하기

```
$ docker run -v ${PWD}/data:/data/db mongo:4
```

3)Client 단에서 접속하기

```
$ ls -al
total 40
drwxr-x--- 5 ubuntu ubuntu 4096 Apr 20 08:08 .
drwxr-xr-x 3 root   root   4096 Apr 19 01:52 ..
-rw------- 1 ubuntu ubuntu  748 Apr 20 05:44 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Jan  6  2022 .bashrc
drwx------ 2 ubuntu ubuntu 4096 Apr 19 01:59 .cache
-rw------- 1 ubuntu ubuntu   20 Apr 20 06:58 .lesshst
-rw-r--r-- 1 ubuntu ubuntu  807 Jan  6  2022 .profile
drwx------ 2 ubuntu ubuntu 4096 Apr 19 01:52 .ssh
-rw-r--r-- 1 ubuntu ubuntu    0 Apr 20 01:35 .sudo_as_admin_successful
drwxr-xr-x 4 lxd    root   4096 Apr 20 08:08 data                    <--- 새로 생성됨.


$ cd ./data
$ ls <----여러개의 파일과 디렉토리 확인
$ docker ps   <--MongoDB PID 확인
예:11b1e9ff12e4

$ sudo docker exec -it PID(앞 2자리도 가능) mongo
MongoDB shell version v4.4.20
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("78a7d009-cbff-4f71-b2a4-7ab9b29ede57") }
MongoDB server version: 4.4.20
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2023-04-20T08:08:40.397+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
        2023-04-20T08:08:41.176+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).
```

```
            The monitoring data will be available on a MongoDB website with a unique URL accessible to you
            and anyone you share the URL with. MongoDB may use this information to make product
            improvements and to suggest MongoDB products and deployment options to you.

            To enable free monitoring, run the following command: db.enableFreeMonitoring()
            To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
    ---
    >
    > show dbs;
    admin   0.000GB
    config  0.000GB
    local   0.000GB


    >use example
    switched to db example
    >db.example.insert({"name":"Henry"})
    WriteResult({"nInserted" : 1})

    >db.example.find({})
    { "_id" : ObjectId("6440f3c3fc7a49aba415d1a2"), "name" : "Henry" }
    >exit
```

$ Server 단에서 Ctrl + C 로 서비스 정지

4)다시 Docker Run을 했을 때 Data가 남아 있을 것인가?

-Server단에서 MongoDB 실행

```
    $ docker run -v ${PWD}/data:/data/db mongo:4
```

-Client 단에서 접속

```
    $ docker ps   <--- PID확인
    예:a63e176204cc

    $ sudo docker exec -it PID(앞 2자리도 가능) mongo
    >show dbs
    admin    0.000GB
    config   0.000GB
    example  0.000GB     <--- example db 확인
    local    0.000GB

    >use example
    >db.example.find({})
    { "_id" : ObjectId("6440f3c3fc7a49aba415d1a2"), "name" : "Henry" }<-- 앞에서 저장한 데이터 확인
```

5)MongoDB Image 모두 삭제

6)다시 Server 단에서 MongoDB Image Run

```
    $ docker run mongo:4
```

7)Client 단에서 접속

```
    $ sudo rm -rf ./data
    $ sudo docker exec -it PID mongo
    >show dbs
    >use example
    >db.example.insert({"name" : "Henry"})
    WriteResult({ "nInserted" : 1 })
    >db.example.find({})
    { "_id" : ObjectId("6440f510ea844ff733e88aa9"), "name" : "Henry" }
    >exit
```

-MongoDB Server도 Ctrl + C로 서비스 정지

8)다시 MongoDB Server Start

```
    $ sudo docker run mongo:4
```

9)Client 단에서 접속

```
    $ sudo docker exec -it PID mongo
    >
    >show dbs
    admin   0.000GB
    config  0.000GB
    local   0.000GB
                  <---example db 없음.
```

6. Container Image 삭제하기

1)Server-side에서 redis 실행하기

```
    $ docker run -p 6379:6379 redis
```

2)Client-side에서

```
    $ sudo apt install redis-tools
    $ redis-cli
    127.0.0.1:6379>set name "Henry"
    OK
    127.0.0.1:6379>get name
    "Henry"
    127.0.0.1:6379>exit

    $ docker ps -a   <-- PID 확인

    $ docker rm PID --> 실패, 이유는 현재 Docker Container 실행 중
    Error response from daemon: You cannot remove a running container a03a7c63ab95fc0fc42d3e85f512c90f7d40200ddcc82aa52d1b4ab9b7c9f332. Stop the container before attempting removal or force remove

    $ sudo docker stop PID  <---클라이언트 세션에서 서버 서비스 중지시킴.
```

3)Container 삭제하기

```
    $ sudo docker ps -a   <--- PID확인
    $ sudo docker rm PID

    $ df
    Filesystem     1K-blocks    Used Available Use% Mounted on
    /dev/root       30297152 4628472  25652296  16% /
    tmpfs             494692       0    494692   0% /dev/shm
    tmpfs             197880     956    196924   1% /run
    tmpfs               5120       0      5120   0% /run/lock
    /dev/xvda15       106858    6182    100677   6% /boot/efi
    tmpfs              98936       4     98932   1% /run/user/1000
```

4)Container Image 삭제하기

```
    # docker images  <--- PID 확인
    # docker rmi PID
```

7. MySQL 사용하기

1)Docker로 MySQL Run

```
    $ mkdir mysql
    $ cd mysql
    $ sudo -i
    # cd /home/ubuntu/mysql
    # docker pull mysql:5.7.34
    # docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=password -d -p 3306:3306 mysql:5.7.34
    # docker ps -a
    CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
    PORTS                                                  NAMES
    80da3367f857   mysql:5.7.34   "docker-entrypoint.s…"   5 seconds ago   Up 4 seconds   0.0.0.0:3306->3306/tcp,
    :::3306->3306/tcp, 33060/tcp   mysql-container
```

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 : ec2-3-39-228-97.ap-northeast-2.compute.amazonaws.com   <--- EC2 Instance Public IPv4 DNS 값
        --Port : 3306                         <--- 미리 보안그룹에서 3306 포트 추가
        --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

```
    $ docker pull httpd

    $ docker images
```

2)Docker Container 구동하기

-docker run 명령을 통해 Container 를 시작하고 Web 서비스를 구성 할 수 있다

```
$ 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.

```
$ docker run -d httpd
$ 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

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

-Port Binding

```
$ docker run -d -p 80:80 httpd
$ 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 exec -it [container ID] bash
/# cd /usr/local/apache2/htdocs
/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