[HOL] Docker Container

1. Docker Hub에서 Container Image 검색하기

1)Docker Version 확인

```
$ docker version
Client: Docker Engine - Community
 Version:           24.0.7
 API version:       1.43
 Go version:        go1.20.10
 Git commit:        afdd53b
 Built:             Thu Oct 26 09:07:41 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          24.0.7
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.10
  Git commit:       311b9ff
  Built:            Thu Oct 26 09:07:41 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.26
  GitCommit:        3dd1e886e55dd695541fdcd67420c2888645a495
 runc:
  Version:          1.1.10
  GitCommit:        v1.1.10-0-g18a0cb0
 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 Tue 2024-01-09 11:15:35 UTC; 2h 56min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2469 (dockerd)
      Tasks: 8
     Memory: 49.9M
        CPU: 14.618s
     CGroup: /system.slice/docker.service
             └─2469 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 09 11:45:12 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:45:12.365228710Z" level=info msg="ignoring event" container=d4dd1fa2c3ce185a6b6193176683dc2ca8948c773cfe0c64>Jan 09 11:46:18 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:46:18.528960385Z" level=error msg="Not continuing with pull after error: manifest unknown: manifest unknown"
Jan 09 11:52:12 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:52:12.133840280Z" level=info msg="ignoring event" container=1c64496bea7d6014b7363112fb81f35771ad7a3c29a052e1>Jan 09 11:52:12 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:52:12.156571470Z" level=warning msg="failed to close stdin: task 1c64496bea7d6014b7363112fb81f35771ad7a3c29a>Jan 09 11:52:46 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:52:46.436605014Z" level=info msg="Container failed to exit within 10s of signal 15 - using the force" contai>Jan 09 11:52:46 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:52:46.481545845Z" level=info msg="ignoring event" container=e961c898643a6ef7d52b5202ab34d70290471c18fb084b6b>Jan 09 11:52:46 ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:52:46.496422609Z" level=warning msg="failed to close stdin: task
```

```
    e961c898643a6ef7d52b5202ab34d70290471c18fb0>Jan 09 11:59:11 ip-172-31-4-124 dockerd[2469]:
    time="2024-01-09T11:59:11.270735509Z" level=info msg="Layer
    sha256:157a4aaa033e0ac8e596a4aa2fadbefe2f6a2a27013294fc145175f99bf64>Jan 09 11:59:11
    ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:59:11.480575062Z" level=info msg="Layer
    sha256:a1360aae5271bbbf575b4057cb4158dbdfbcae76698189b55fb1039bc0207>Jan 09 11:59:11
    ip-172-31-4-124 dockerd[2469]: time="2024-01-09T11:59:11.502627513Z" level=info msg="Layer
    sha256:ac28800ec8bb38d5c35b49d45a6ac4777544941199075dff8c4eb63e093aa>
```

3)Docker Hub에서 nginx 검색하기

```
    $ docker search nginx
    NAME                               DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
    nginx                              Official build of Nginx.                        19452     [OK]
    unit                               Official build of NGINX Unit: Universal Web …   20        [OK]
    nginxinc/nginx-unprivileged        Unprivileged NGINX Dockerfiles                  140
    nginx/nginx-ingress                NGINX and  NGINX Plus Ingress Controllers fo…   87
    nginx/nginx-prometheus-exporter    NGINX Prometheus Exporter for NGINX and NGIN…   33
    nginxinc/nginx-s3-gateway          Authenticating and caching gateway based on …   3
    nginx/unit                         This repository is retired, use the Docker o…   64
    nginx/nginx-ingress-operator       NGINX Ingress Operator for NGINX and NGINX P…   2
    nginxinc/amplify-agent             NGINX Amplify Agent docker repository           1
    nginx/nginx-quic-qns               NGINX QUIC interop                              1
    nginxinc/ingress-demo              Ingress Demo                                    4
    nginxproxy/nginx-proxy             Automated nginx proxy for Docker containers …   124
    nginxproxy/acme-companion          Automated ACME SSL certificate generation fo…   128
    bitnami/nginx                      Bitnami nginx Docker Image                      181                  [OK]
    bitnami/nginx-ingress-controller   Bitnami Docker Image for NGINX Ingress Contr…   32                   [OK]
    ubuntu/nginx                       Nginx, a high-performance reverse proxy & we…   104
    nginxinc/nginmesh_proxy_debug                                                      0
    nginxproxy/docker-gen              Generate files from docker container meta-da…   14
    nginxinc/mra-fakes3                                                                0
    kasmweb/nginx                      An Nginx image based off nginx:alpine and in…   6
    nginxinc/ngx-rust-tool                                                             0
    nginxinc/mra_python_base                                                           0
    rancher/nginx-ingress-controller                                                   11
    nginxinc/nginmesh_proxy_init                                                       0
```

2. Container Image 다운로드 후 Image Layer 보기

1)다음 명령을 통해 /var/lib/docker/overlay2로 이동하여 'l' 디렉토리를 제외한 모든 디렉토리 삭제되어 있음을 확인

```
    $ docker system prune
    $ docker volume prune
    $ sudo -i
    # cd /var/lib/docker/overlay2
```

```
2)# ls -l
total 4
drwx------ 2 root root 4096 Jan  9 12:00 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 Jan  9 11:15 buildkit
drwx--x---  2 root root 4096 Jan  9 11:59 containers
-rw-------  1 root root   36 Jan  9 11:15 engine-id
drwx------  3 root root 4096 Jan  9 11:15 image
drwxr-x---  3 root root 4096 Jan  9 11:15 network
drwx--x--- 10 root root 4096 Jan  9 14:16 overlay2
drwx------  4 root root 4096 Jan  9 11:15 plugins
drwx------  2 root root 4096 Jan  9 11:15 runtimes
drwx------  2 root root 4096 Jan  9 11:15 swarm
drwx------  2 root root 4096 Jan  9 14:16 tmp
drwx-----x  2 root root 4096 Jan  9 11:15 volumes

# cd overlay2      <---7개의 directory 확인
# ls -l
total 32
drwx--x--- 4 root root 4096 Jan  9 14:16 27c85982bc00c3a56846be27d935ade140bb1b27a1535b0cdb4f5d73757d3bd6
drwx--x--- 3 root root 4096 Jan  9 14:16 3dd12fe37da7b01668a7ff40264ca9dbcc1f28966ea703289d3606a9c4ab5cec
drwx--x--- 4 root root 4096 Jan  9 14:16 8763a5d715a46fa7cdcb431350802599b433e74e8a01fb9c2a7b68a74bcf187f
drwx--x--- 4 root root 4096 Jan  9 14:16 a4d5ee4ccf714f224f903bcde0facfbf76e11fa3c19b6e52929b779795143e14
drwx--x--- 4 root root 4096 Jan  9 14:16 dc1bcaa4e7c746bcfee8f707f8336bcc314e113e80819e60d5fb6a94f7839a97
drwx--x--- 4 root root 4096 Jan  9 14:16 f74132eabff27e5d3ad7353145149d8cdabf173fc4d506c5eef09a4033df3971
drwx--x--- 4 root root 4096 Jan  9 14:16 fc779983d3c5ab9e9b3d976c3b2251ab1346d4303e513a65df84b88d9d649e62
drwx------ 2 root root 4096 Jan  9 14:16 l


# cd /home/ubuntu
# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    d453dd892d93   2 months ago   187MB
```

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    d453dd892d93   2 months ago   187MB
```

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
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
PORTS                                 NAMES
d9fa05a924fd   nginx:latest   "/docker-entrypoint.…"   33 seconds ago   Up 32 seconds   0.0.0.0:80->80/tcp,
:::80->80/tcp   webserver

# docker stop werserver
webserver

# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS
PORTS     NAMES
d9fa05a924fd   nginx:latest   "/docker-entrypoint.…"   About a minute ago   Exited (0) 15 seconds
ago             webserver
```

4)docker Container remove

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

5)docker Image remove

```
# docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
nginx        latest    d453dd892d93   2 months ago   187MB

# docker rmi nginx
Error response from daemon: conflict: unable to remove repository reference "nginx" (must force) - container
d9fa05a924fd is using its referenced image d453dd892d93

# docker rmi -f nginx
Untagged: nginx:latest
Untagged: nginx@sha256:2bdc49f2f8ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026
Deleted: sha256:d453dd892d9357f3559b967478ae9cbc417b52de66b53142f6c16c8a275486b9

# docker image ls
```

```
EPOSITORY    TAG        IMAGE ID    CREATED    SIZE

# ls -l /var/lib/docker/overlay2/
total 40
drwx--x--- 4 root root 4096 Jan  9 14:16 27c85982bc00c3a56846be27d935ade140bb1b27a1535b0cdb4f5d73757d3bd6
drwx--x--- 3 root root 4096 Jan  9 14:16 3dd12fe37da7b01668a7ff40264ca9dbcc1f28966ea703289d3606a9c4ab5cec
drwx--x--- 4 root root 4096 Jan  9 14:21 596bb185ba2787343a69b806d32c3b4b849a8db4411f98a6cd2c627fde1f0143
drwx--x--- 4 root root 4096 Jan  9 14:20 596bb185ba2787343a69b806d32c3b4b849a8db4411f98a6cd2c627fde1f0143-init
drwx--x--- 4 root root 4096 Jan  9 14:16 8763a5d715a46fa7cdcb431350802599b433e74e8a01fb9c2a7b68a74bcf187f
drwx--x--- 4 root root 4096 Jan  9 14:16 a4d5ee4ccf714f224f903bcde0facfbf76e11fa3c19b6e52929b779795143e14
drwx--x--- 4 root root 4096 Jan  9 14:16 dc1bcaa4e7c746bcfee8f707f8336bcc314e113e80819e60d5fb6a94f7839a97
drwx--x--- 4 root root 4096 Jan  9 14:16 f74132eabff27e5d3ad7353145149d8cdabf173fc4d506c5eef09a4033df3971
drwx--x--- 4 root root 4096 Jan  9 14:20 fc779983d3c5ab9e9b3d976c3b2251ab1346d4303e513a65df84b88d9d649e62
drwx------ 2 root root 4096 Jan  9 14:20 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
af107e978371: Already exists
336ba1f05c3e: Already exists
8c37d2ff6efa: Already exists
51d6357098de: Already exists
782f1ecce57d: Already exists
5e99d351b073: Already exists
7b73345df136: Already exists
Digest: sha256:2bdc49f2f8ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026
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: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/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
2024/01/09 14:25:56 [notice] 1#1: using the "epoll" event method
2024/01/09 14:25:56 [notice] 1#1: nginx/1.25.3
2024/01/09 14:25:56 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/01/09 14:25:56 [notice] 1#1: OS: Linux 6.2.0-1017-aws
2024/01/09 14:25:56 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/01/09 14:25:56 [notice] 1#1: start worker processes
2024/01/09 14:25:56 [notice] 1#1: start worker process 30
                                <--- log 대기중
```

2)Client-side에서   <---다른 세션으로 연결하여

```
$ 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   <---서버쪽에서 확인

```
    172.17.0.1 - - [09/Jan/2024:14:27:05 +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

```
    172.17.0.1 - - [09/Jan/2024:14:31:52 +0000] "GET /aaa.html HTTP/1.1" 404 153 "-" "curl/7.81.0" "-"
    2024/01/09 14:31:52 [error] 30#30: *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"
```

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

```
    ^C2024/01/09 14:32:34 [notice] 1#1: signal 2 (SIGINT) received, exiting
    2024/01/09 14:32:34 [notice] 30#30: exiting
    2024/01/09 14:32:34 [notice] 30#30: exit
    2024/01/09 14:32:34 [notice] 1#1: signal 17 (SIGCHLD) received from 30
    2024/01/09 14:32:34 [notice] 1#1: worker process 30 exited with code 0
    2024/01/09 14:32:34 [notice] 1#1: exit
```

-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
        root@ip-172-31-4-124:/home/ubuntu# 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: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/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
2024/01/09 14:34:23 [notice] 1#1: using the "epoll" event method
2024/01/09 14:34:23 [notice] 1#1: nginx/1.25.3
2024/01/09 14:34:23 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/01/09 14:34:23 [notice] 1#1: OS: Linux 6.2.0-1017-aws
2024/01/09 14:34:23 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/01/09 14:34:23 [notice] 1#1: start worker processes
2024/01/09 14:34:23 [notice] 1#1: start worker process 30
                         <----- 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. Docker Volume Mount하기

1)Server 단에서 MongoDB search

```
$ docker search mongodb
NAME                                              DESCRIPTION                                      STARS     OFFICIAL
AUTOMATED
mongo                                             MongoDB document databases provide high avai…   10031
[OK]
mongo-express                                     Web-based MongoDB admin interface, written w…   1392
[OK]
mongodb/mongodb-atlas-kubernetes-operator         The MongoDB Atlas Kubernetes Operator -
Kube…   5
mongodb/mongodb-community-server                  The Official MongoDB Community Server
62
mongodb/mongodb-enterprise-server                 The Official MongoDB Enterprise Advanced Ser…
6
mongodb/mongodb-atlas-kubernetes-operator-prerelease   This is an internal-use-only build of the Mo…
```

```
    0
    bitnami/mongodb                              Bitnami MongoDB Docker Image
    238             [OK]
    mongodb/signatures                           Signatures for container images             0
    mongodb/atlas                               Create, manage, and automate MongoDB Atlas r...
    4
    mongodb/mongodb-atlas-search                                                               0
    bitnami/mongodb-exporter                                                                12
    bitnami/mongodb-sharded                                                                 11
    mongodb/apix_test                            apix test repo                             0
    rapidfort/mongodb                            RapidFort optimized, hardened image for Mong...
    23
    rapidfort/mongodb-ib                         RapidFort optimized, hardened image for Mong...
    10
    rapidfort/mongodb-official                   RapidFort optimized, hardened image for Mong...
    11
    rancher/mongodb-conf                          This container image is no longer maintained...
    2
    bitnamicharts/mongodb                                                                   0
    mongodb/mongo-cxx-driver                        Container image for the C++ driver
    2
    rapidfort/mongodb-perfomance-test                                                          10
    bitnamicharts/mongodb-sharded                                                             0
    rancher/mongodb-config                                                                  0
    drud/mongodb                                 Mongodb                                      0              [OK]
    dockette/adminer                            My most tiniest (10mb) Adminer image with su...
    20              [OK]
    edgexfoundry/docker-edgex-mongo                    ARCHIVED! MongoDB container for older versio...
    5
```

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
bye

$ 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