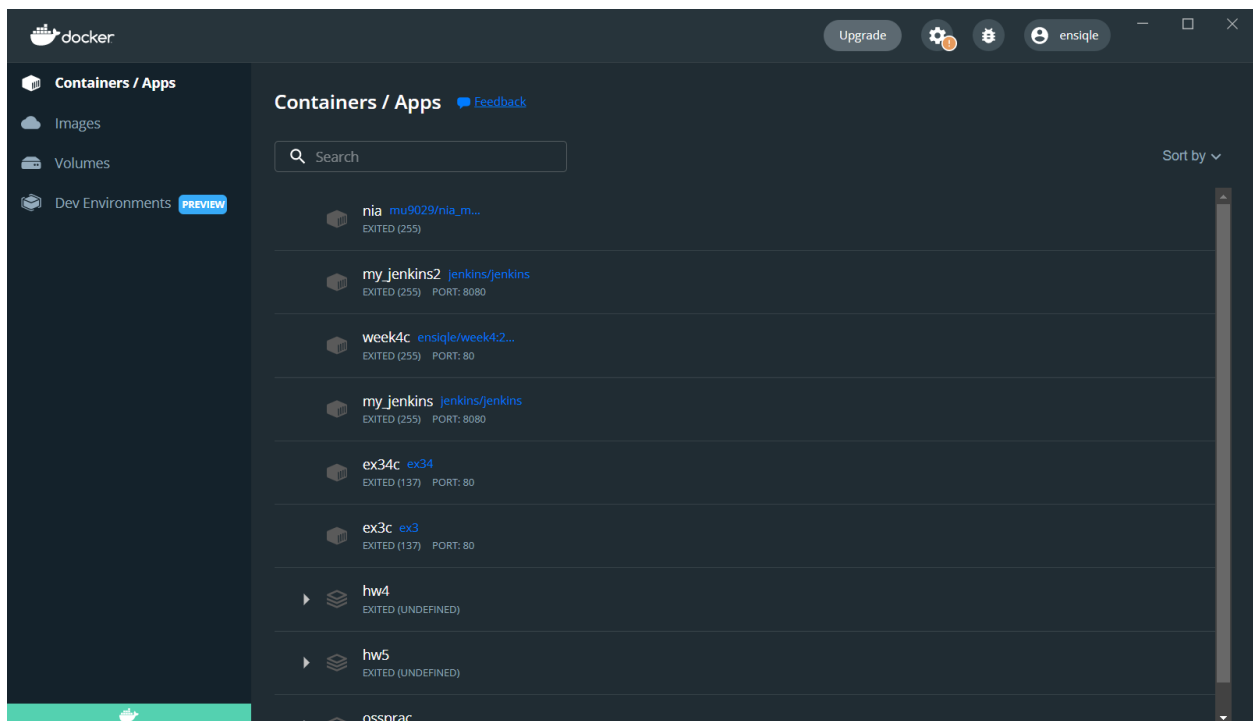


9주차(2)_Docker 실습

ex0) Docker 실행

Docker 실행

- **Window** : 시작버튼 옆 검색으로 docker Desktop을 실행하면 우측 하단 상태 표시줄에 docker가 뜹니다. 우클릭으로 누른 후 Dashboard를 눌러주세요.
- **Mac** : spotlight에 docker를 검색하여 실행해줍니다. 이후에 우측 상단에 docker 아이콘이 나타나면 우클릭으로 누른 후 Dashboard를 눌러서 실행할 수도 있습니다.



terminal

cmd / terminal에 **docker** 를 입력하고 docker COMMAND usage가 뜨면 설치가 완료된 것입니다.

```

PS C:\Users\kimmi> docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

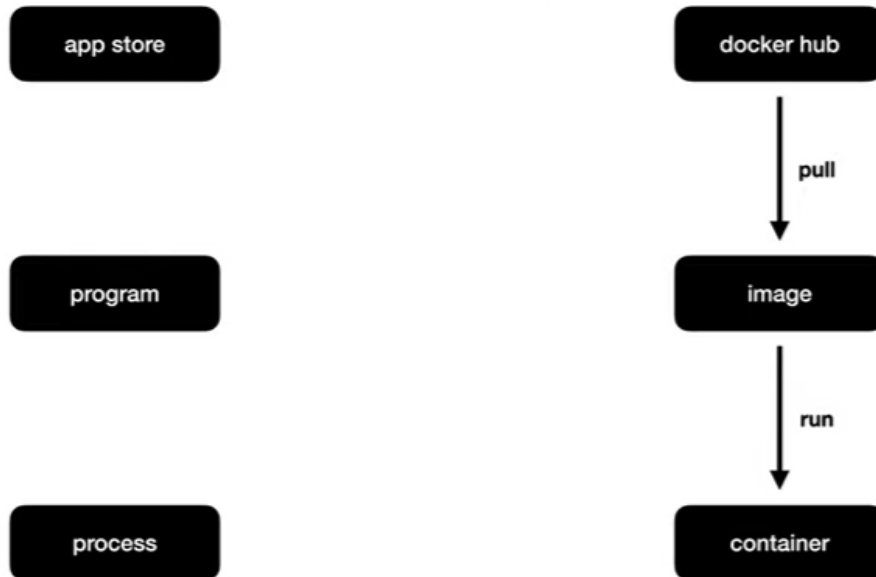
Options:
  --config string      Location of client config files (default
                        "C:\\Users\\kimmi\\.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\\Users\\kimmi\\.docker\\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\\Users\\kimmi\\.docker\\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\\Users\\kimmi\\.docker\\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder      Manage builds
  buildx       Docker Buildx (Docker Inc., v0.8.2)

```

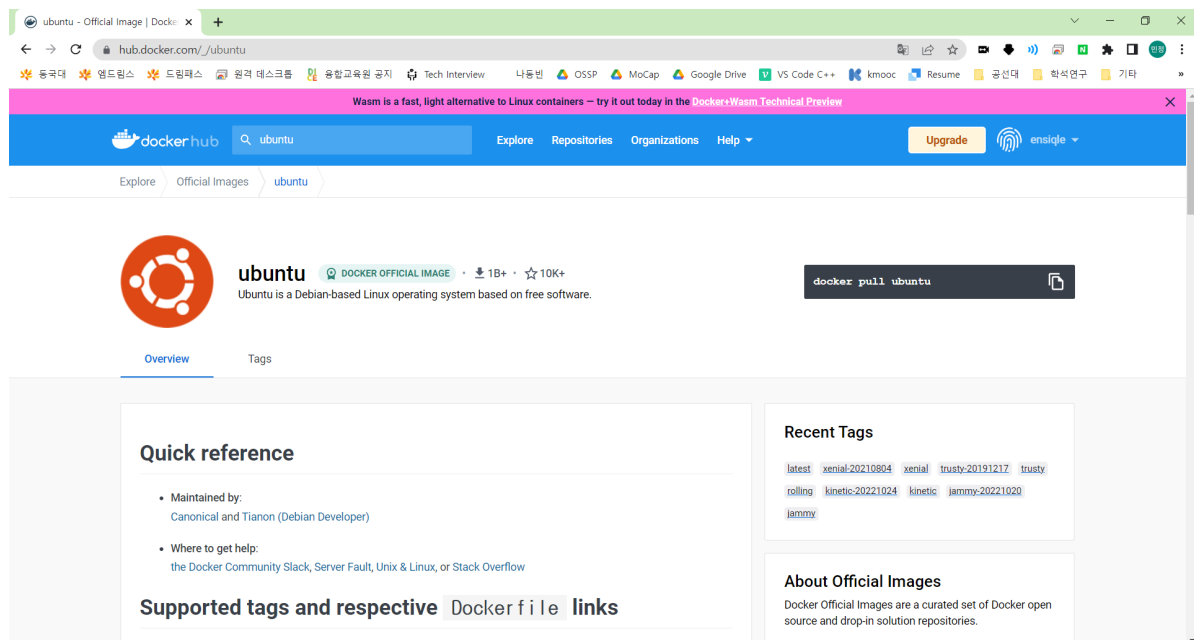
ex1) 이미지 다운로드

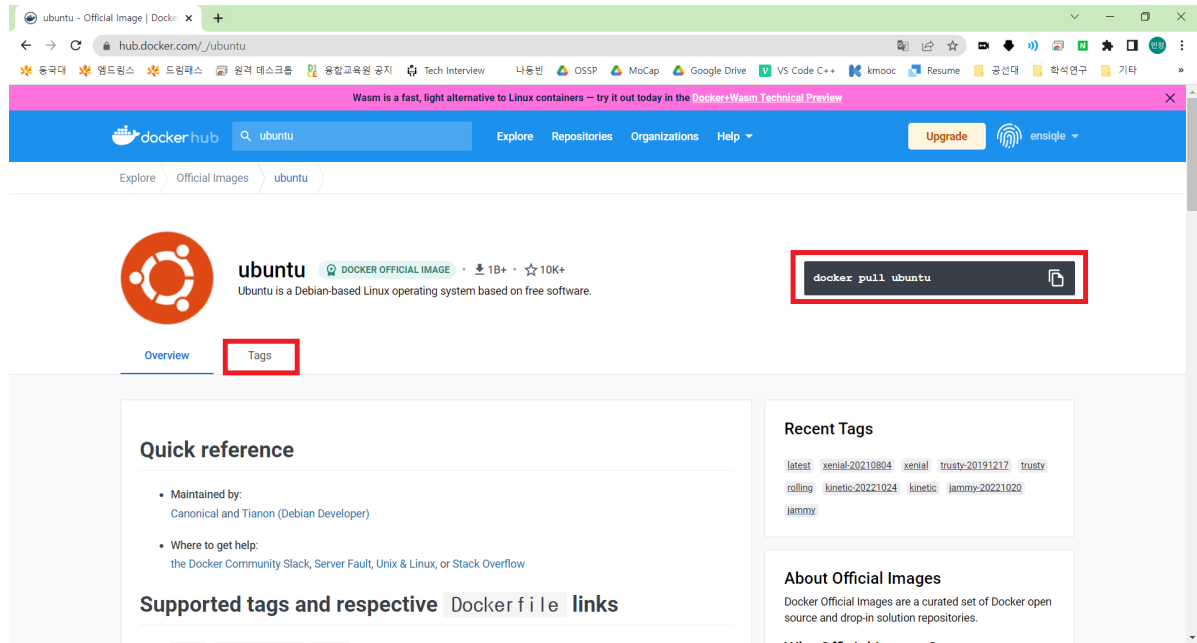
필요한 image를 다운받는 실습입니다.



1. <https://hub.docker.com/> 접속 후 로그인해주세요.

2. Ubuntu를 검색해주세요.





3. 이미지 다운로드

아래의 명령어를 사용하여 우분투 이미지를 pull 받아주세요.

태그명을 생략할 때도, 가장 최신 버전을 자동으로 내려받습니다!

```
docker pull [옵션] <이미지명>[:태그명]
```

```
// 예시
docker pull ubuntu:latest
docker pull ubuntu
```

```
PS C:\Users\kimmi> docker pull ubuntu:latest
latest: Pulling from library/ubuntu
301a8b74f71f: Pull complete
Digest: sha256:7cfe75438fc77c9d7235ae502bf229b15ca86647ac01c844b272b56326d56184
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

4. 다운받은 이미지 확인

```
docker image
```

```
PS C:\Users\kimmi> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu               latest             cdb68b455a14       33 hours ago       77.8MB
hrmark1010/ex4      2.0               1410+dce7c42       4 months ago       915MB
hrmark1010/ex4      6                 6daa7d4eddde       4 months ago       915MB
ensiqle/week4       2.0               737adab76186       4 months ago       915MB
ensiqle/week4       latest            737adab76186       4 months ago       915MB
jenkins/jenkins     latest            011729edd836       5 months ago       460MB
mu9029/nia_model    1.4               20bca428e2f5       8 months ago       5.9GB
```



Tag란?

Image의 버전이라고 생각하시면 됩니다! 앞에서 `docker pull ubuntu:latest` 라는 명령어를 사용하여 이미지를 pull받았는데, 이때 latest가 tag입니다.

`docker pull ubuntu:xenial-20210804` 을 사용하면 다른 tag의 이미지를 pull 받을 수 있습니다.

The screenshot shows the Docker Hub interface for the 'ubuntu' repository. The 'Tags' tab is selected, and the tags are sorted by 'Newest'. The 'latest' tag is highlighted with a red box. Below it, the 'xenial-20210804' tag is also highlighted with a red box. The page shows details for each tag, including the digest, OS/ARCH, and compressed size.

TAG	DIGEST	OS/ARCH	COMPRESSED SIZE
latest	ddb688d8d15	linux/amd64	29.02 MB
	cb6ce0db39ad	linux/arm/v7	25.77 MB
	ae64a209ba12	linux/arm64/v8	27.07 MB
	+3 more...		
xenial-20210804	bcb8397f1390	linux/386	43.7 MB
	a3785f78ab85	linux/amd64	44.35 MB
	5c33df0beb44	linux/arm/v7	38.45 MB
	+3 more...		

ex2) 컨테이너 생성 및 실행

1. 컨테이너 생성

```
docker run [옵션] <이미지명> [명령]

// 예시
docker run -dit --name my-ubuntu ubuntu:latest
```

```
PS C:\Users\kimmi> docker run -dit --name my-ubuntu ubuntu:latest
3fb2757f74906ad0fc1ab52c54c9800c4ecd4abb1140bd98022c6344dabb8ada
```

- 옵션

`-i` : interactive 모드. bash(command line)에 명령어를 입력할 수 있게 함

`-t` : tty모드이며 bash를 사용하기 위해서는 해당 옵션을 설정해야 함, 이 옵션을 설정하지 않으면 명령을 입력할 수 는 있지만 shell이 표시되지 않음!

`-d` : detached 모드 : 데몬모드이며 백그라운드로 실행됨

`--name` : 컨테이너의 이름을 설정함

2. 도커 컨테이너 목록 조회

```
docker ps
또는
docker ps -a
```

```
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE          COMMAND         CREATED        STATUS        PORTS          NAMES
3fb2757f7490   ubuntu:latest  "bash"         17 hours ago   Up 17 hours   -              my-ubuntu
```

`docker ps` : 현재 실행되고 있는 도커 컨테이너의 목록을 볼 수 있음

`docker ps -a` : 종료된 컨테이너를 포함해서 모든 컨테이너 목록을 보고 싶을 때! exit 명령어로 빠져나왔다면, 컨테이너가 종료되었기 때문에 실행되고 있는 컨테이너가 보이지 않았을 것!

ex3) 컨테이너에서 명령어 실행

1. 컨테이너 안으로 들어가기

```
docker exec -it <컨테이너 이름> /bin/bash
```

// 예시

```
docker exec -it my-ubuntu /bin/bash
```

```
PS C:\Users\kimmi> docker exec -it my-ubuntu /bin/bash
root@3fb2757f7490:/# |
```

2. 명령어 실행

```
root@3fb2757f7490:/# pwd
/
root@3fb2757f7490:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@3fb2757f7490:/# |
```

3. 나가기

```
exit
```

```
root@3fb2757f7490:/# exit
exit
PS C:\Users\kimmi> |
```

ex4) 컨테이너 및 이미지 삭제

1. 컨테이너 중지

```
docker stop my-ubuntu
```

```
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
3fb2757f7490   ubuntu:latest  "bash"    17 hours ago    Up 7 seconds
PS C:\Users\kimmi> docker stop my-ubuntu
my-ubuntu
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\kimmi>
```

2. 컨테이너 다시 실행

```
docker restart my-ubuntu
```

```
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\kimmi> docker restart my-ubuntu
my-ubuntu
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
3fb2757f7490   ubuntu:latest  "bash"    17 hours ago    Up 2 seconds
PS C:\Users\kimmi>
```

3. 컨테이너 삭제

컨테이너가 실행중이면 **불가능**

```
docker stop my-ubuntu    // 컨테이너 실행 종료
docker rm my-ubuntu      // 컨테이너 삭제
```

- 실행 종료하지 않고 삭제했을 때

```
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
3fb2757f7490   ubuntu:latest  "bash"    17 hours ago    Up 5 minutes
PS C:\Users\kimmi> docker rm my-ubuntu
Error response from daemon: You cannot remove a running container 3fb2757f74906ad0fc1ab52c54c44dabb8ada. Stop the container before attempting removal or force remove
PS C:\Users\kimmi>
```

- 실행 종료 후 삭제했을 때


```
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
3fb2757f7490   ubuntu:latest  "bash"                  17 hours ago   Up 7 minutes                my-ubuntu
PS C:\Users\kimmi> docker stop my-ubuntu
my-ubuntu
PS C:\Users\kimmi> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
PS C:\Users\kimmi> docker rm my-ubuntu
my-ubuntu
```

4. 이미지 삭제

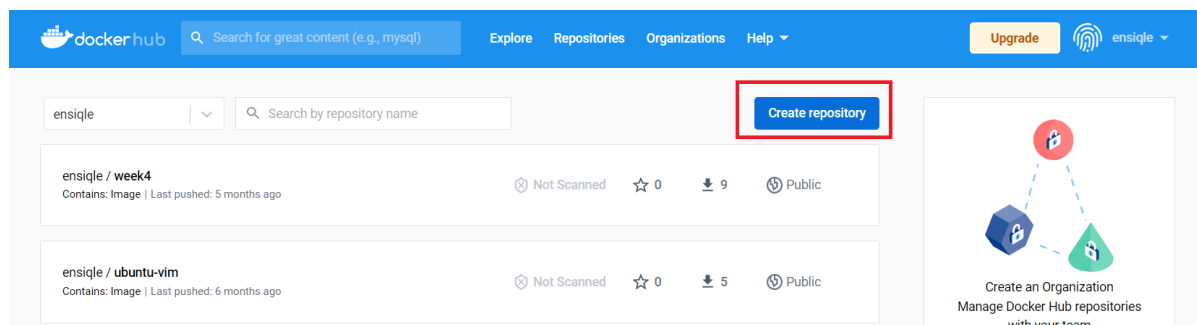
```
docker rmi ubuntu
```

```
PS C:\Users\kimmi> docker rmi ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:7cfe75438fc77c9d7235ae502bf229b15ca86647ac01c844b272b56326d56184
Deleted: sha256:cdb68b455a141ed921945f6d39a8c0694a7e21a37b2b030488d73e38875a26cc
Deleted: sha256:7ea4455e747ead87d6cc1c4efaf3a79530a931a0856a9f9ce9ac2d8d45bd3c28
PS C:\Users\kimmi>
```

ex5) Docker hub에 Docker Image 업로드

1. Commit

docker hub에서 repository를 생성해주세요.



Create repository

ensiqle

▼

oosp

Description

Visibility

Using 0 of 1 private repositories. [Get more](#)

☒ Public

☐ Private

Appears in Docker Hub search results

Only visible to you

Cancel

Create

(필수X) 먼저 기존에 설치한 컨테이너를 실행 후 업로드할 내용을 변경 해줍니다. 저는 my-ubuntu 컨테이너에 apt update만 수행하였습니다.

```
docker exec -it my-ubuntu /bin/bash
```

```
PS C:\Users\kimmi> docker exec -it my-ubuntu /bin/bash
root@c01901b1ba98:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [577 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
0% [4 InRelease 95.9 kB/114 kB 84%] [3 Packages 217 kB/577 kB 38%]
0% [4 InRelease 97.2 kB/114 kB 85%] [3 Packages 219 kB/577 kB 38%]
```

이후 컨테이너 실행 종료 후, oosp라는 레포지토리에 1.0이라는 태그로 커밋을 진행하였습니다.

```
docker commit my-ubuntu [본인아이디]/[레포지토리명]:[tag]

// 예시
docker commit my-ubuntu ensiqle/oosp:1.0
```

```

root@c01901b1ba98:/# exit
exit
PS C:\Users\kimmi> docker commit my-ubuntu ensiqle/ossip:1.0
sha256:f5e1130a8391f65881fc3aa7081e70f239d357fc9a121e28f92d3bd2eb3b90c7
PS C:\Users\kimmi>

```

2. 로그인

아래의 명령어를 사용하여 docker에 로그인 해줍니다. 저는 현재 로그인이 되어 있어서 자동 로그인이 되었지만, 여러분들은 직접 username과 password를 입력할 수도 있습니다!

```
docker login
```

```

PS C:\Users\kimmi> docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn
more: https://docs.docker.com/go/access-tokens/
PS C:\Users\kimmi>

```

3. push

```
docker push [본인아이디]/[repository이름]:[tag]
```


```
// 예시
```

```
docker push ensiqle/ossip:1.0
```

```

PS C:\Users\kimmi> docker push ensiqle/ossip:1.0
The push refers to repository [docker.io/ensiqle/ossip]
96200161a395: Pushed
7ea4455e747e: Mounted from library/ubuntu
1.0: digest: sha256:cc921ebdf258e0e197031049e9c35eb8c094b57892c317b5752ab440de74413b size: 740
PS C:\Users\kimmi>

```

 ensiqle / ossp

Description

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

Public View

To push a new tag to this repository,


docker push ensiqle/ossps:tagname

Tags and scans

VULNERABILITY SCANNING - DISABLED

Enable

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
<div><div></div>1.0</div>		Image	---	a few seconds ago

[See all](#)

[Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

Upgrade

[Learn more](#)