



TABA:Docker 실습

2023.08.03

전성환

72160261@dankook.ac.kr

도커

■ Docker

- 2013년 3월 dotCloud 창업자 Solomon Hykes가 Pycon Conference에서 발표
- 애플리케이션을 컨테이너로 좀더 쉽게 사용할 수 있게 만든 오픈소스 프로젝트
- 격리된 가상공간을 만들긴 하지만 실제 수행을 호스트 운영 체제에서 수행
- 가상머신과 달리 성능손실이 거의 없는 차세대 클라우드 솔루션으로 주목

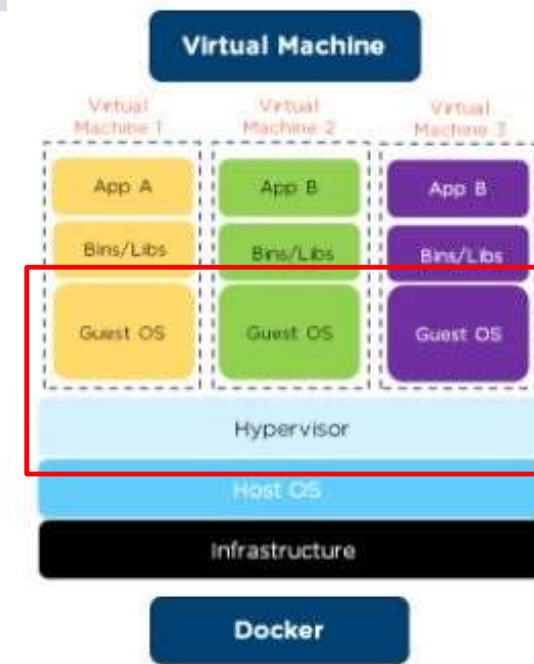
■ Docker Project

- Docker Compose, Docker Machine, Registry 등 다양한 프로젝트 존재
- 일반적 도커 : Docker Engine을 의미
- Docker Project는 Docker Engine을 효율적으로 사용하기 위한 도구들.

가상화 vs 도커

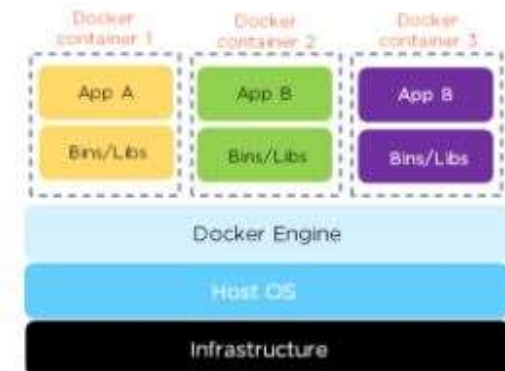
■ VM (Virtual Machine)

- 구동중인 운영체제와 각종 라이브러리까지 전체 영역을 독립적으로 구분
- 하이퍼바이저를 통한 가상화로 성능 손실이 발생
- 실제 앱이 구동되기까지 시간이 오래 걸림.



■ Docker

- 애플리케이션 실행에 필요한 라이브러리만 필요하기 때문에 적은 리소스 사용.
- 애플리케이션 부분만 가상화 되어 실행되기 때문에 성능이 빠르다.



도커 구조

■ Docker Client

- docker run과 같은 명령어를 API로 변환해서 dockerd에 요청

■ Docker Daemon(dockerd)

- 도커 이미지 관리, 이미지 빌드 등 다양한 작업 수행
- containerd 호출

■ containerd

- docker에서 분리되어 오픈소스로 운영되고 있는 컨테이너 런타임
- Container Runtime Interface(CRI)

■ runc

- 실제 컨테이너를 만드는 곳.
- Low-level Container Runtime

Docker Client

CLI, Compose ...

```
$ docker container run ...
```

Docker API - TLS/Unix Socket
POST /containers/create HTTP 1.1

Docker Daemon

Docker API
Log Management
Storage Management
libnetwork
SwarmKit
BuildKit
Docker Content Trust
Image Management



gRPC Call
client.NewContainer(context, ...)

High-Level Runtime

containerd

containerd

shim

shim

shim

shim

OCI Layer

Low-Level Runtime
runc

runc

runc

runc

containers



도커 설치

- 도커 설치 참조

- <https://docs.docker.com/engine/installation/#time-based-release-schedule>

- 실습 환경

- Centos Stream release 9

- yum utils 설치

- yum install yum-utils

- 최신 도커엔진 설치용 리포지터리 추가

- `sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo`

도커 설치

■ 도커 엔진 설치

□ sudo yum install docker-ce

■ 도커 엔진 시작

□ sudo systemctl start docker

■ 도커 엔진 테스트

□ docker run hello-world

```
[shjeon@localhost ~]$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:926fac19d22aa2d60fla276b66a20eb765fbbea2db5dbdaafeb456ad8ce81598
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

```
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
```

```
For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

도커 이미지

■ 도커 이미지

- 가상머신 생성시 사용하는 **ISO** 와 비슷한 개념
- 여러 개의 층으로 된 바이너리 파일로 존재
- 컨테이너 생성시 읽기 전용으로 사용됨
- 도커 명령어로 레지스트리(Docker hub)로부터 다운로드 가능

TEST/ubuntu:20.04

저장소 이름

이미지 이름

이미지 버전

- 저장소 이름: 이미지가 저장된 장소(생략 시 Docker hub)
- 이미지 이름: 이미지의 역할을 나타낼 이름 (생략 불가능)
- 이미지 버전: 이미지 버전정보(생략 시 latest)

도커 컨테이너 다루기

■ 도커 그룹에 유저 추가하기

- `sudo groupadd docker` (docker group 생성)
- `sudo usermod -aG docker $USER` (docker group 해당 유저 추가)
- `newgrp docker` (적용하기)

■ 도커 엔진 버전 확인

- `docker -v`

```
[shjeon@localhost ~]$ docker -v
Docker version 24.0.5, build ced0996
```


도커 컨테이너 다루기

■ 컨테이너 생성

- `docker run -i -t ubuntu:20.04`

```
[shjeon@localhost ~]$ docker run -i -t ubuntu:20.04
Unable to find image 'ubuntu:20.04' locally
20.04: Pulling from library/ubuntu
01085d60b3a6: Pull complete
Digest: sha256:c9820a44b950956a790c354700c1166a7ec648bc0d215fa438d3a339812f1d01
Status: Downloaded newer image for ubuntu:20.04
```

- `run` : 컨테이너 실행
- `-i` : 컨테이너와 상호 입출력 가능 옵션
- `-t` : 셸을 사용할 수 있는 `tty` 활성화

도커 컨테이너 다루기

- 컨테이너 파일 시스템 확인 및 호스트 네임 확인

- ls
- hostname

```
root@83393ccad578:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@83393ccad578:/# hostname
83393ccad578
```

- 컨테이너 내부에서 빠져나오기

- **exit** (컨테이너 종료 후 빠져나오기)
- **Ctrl+P, Ctrl+Q** 차례대로 입력 (컨테이너를 종료하지 않고 빠져나오기)

도커 컨테이너 다루기

- 이미지 내려받기
 - `docker pull centos`

```
[shjeon@localhost ~]$ docker pull centos
Using default tag: latest
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
```

- 이미지 목록 확인
 - `docker search centos`

```
[shjeon@localhost ~]$ docker images
REPOSITORY      TAG           IMAGE ID       CREATED        SIZE
ubuntu          20.04        14be0685b768   5 weeks ago    72.8MB
hello-world     latest       9c7a54a9a43c   2 months ago   13.3kB
centos          latest       5d0da3dc9764   22 months ago  231MB
```

도커 컨테이너 다루기

■ 컨테이너 생성하기

- `docker create -i -t --name mylinux centos`

```
[shjeon@localhost ~]$ docker create -i -t --name mylinux centos
f8b5e6452372f4b71432de1f28bf9d92b745331890f902a35a8fcaa8bf41ae41
```

- `create` : 컨테이너 생성(생성 후 실행 x)
- `--name` : 컨테이너 이름 지정

■ 컨테이너 목록 확인

- `docker ps`

```
[shjeon@localhost ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
83393ccad578	ubuntu:20.04	"/bin/bash"	16 minutes ago	Up 16 minutes		sad_murdock

도커 컨테이너 다루기

■ 컨테이너 목록 확인

- `docker ps` (현재 실행중인 목록 출력)
- `docker ps -a` (모든 컨테이너 목록 출력)

```
[shjeon@localhost ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f8b5e6452372	centos	"/bin/bash"	9 minutes ago	Exited (127) 52 seconds ago		mylinux
83393ccad578	ubuntu:20.04	"/bin/bash"	24 minutes ago	Up 24 minutes		sad_murdock
298f72f00ce1	ubuntu:20.04	"/bin/bash"	24 minutes ago	Exited (0) 24 minutes ago		loving_haibt
a767646d4d9b	hello-world	"/hello"	5 hours ago	Exited (0) 5 hours ago		modest_shtern

- **IMAGE** : 컨테이너 생성시 사용된 이미지 이름
- **COMMAND** : 컨테이너 시작시 실행될 명령어 (기본 내장된 명령어 `/bin/bash`)
- **CREATED** : 컨테이너가 생성된 이후 시간
- **STATUS** : 컨테이너 상태 (**UP** : 실행중, **Exited** : 중지됨, **Pause** : 일시중지)
- **PORTS** : 컨테이너가 오픈한 포트와 호스트에 연결 상태
- **NAMES** : 컨테이너 고유 이름, 중복 불가능, 변경가능

도커 컨테이너 다루기

■ 컨테이너 삭제

- `docker rm` 컨테이너 이름
- `docker rm mylinux`

```
[shjeon@localhost ~]$ docker rm mylinux
mylinux
[shjeon@localhost ~]$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
83393ccad578	ubuntu:20.04	"/bin/bash"	28 minutes ago	Up 28 minutes		sad_murdock
298f72f00ce1	ubuntu:20.04	"/bin/bash"	28 minutes ago	Exited (0) 28 minutes ago		loving_haibt
a767646d4d9b	hello-world	"/hello"	5 hours ago	Exited (0) 5 hours ago		modest_shtern

- `rm` : 컨테이너 삭제 (단, 중지된 컨테이너만 삭제 가능)
- `rm -f` : 실행중인 컨테이너 삭제
- `stop` : 컨테이너 중지

도커 컨테이너 다루기

- 중지된 모든 컨테이너 삭제

- docker container prune

```
[shjeon@localhost ~]$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
298f72f00ce15f58290c4664da2a4b2b88745cd8718433b6f180222a9a83488a
a767646d4d9bbd443b9d648f31e76cd36606cbaa442b5f882ff2d139bf18e169

Total reclaimed space: 24B
```

- 모든 컨테이너 정지 및 삭제

- docker stop \$(docker ps -a -q)
 - docker rm \$(docker ps -a -q)
 - ps -q : 상태 관계없이 모든 컨테이너 출력
 - ps -a : 컨테이너 id만 출력

컨테이너 애플리케이션 만들기

■ 서비스 컨테이너 화

- 컨테이너에 하나의 애플리케이션만 실행
- 컨테이너간 독립성 보장
- 한 컨테이너에 프로세스 하나만 실행

■ 워드프레스 서비스 구축

- 데이터베이스와 워드프레스 웹서버 컨테이너 생성 및 연동
- MySQL 컨테이너 생성 및 실행
- `docker run --name wordpressdb -e MYSQL_ROOT_PASSWORD=test1 -d mysql`

```
[shjeon@localhost ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
83fadb541c22	mysql	"docker-entrypoint.s..."	2 seconds ago	Up 1 second	3306/tcp, 33060/tcp	wordpressdb

컨테이너 애플리케이션 만들기

■ 생성된 MySQL 테스트

- docker exec -it wordpressdb bash

```
[shjeon@localhost ~]$ docker exec -it wordpressdb bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- mysql -u root -p

- 패스워드 입력 후 MySQL 실행상태 확인

컨테이너 애플리케이션 만들기

■ 워드프레스 구축

□ Wordpress 웹 컨테이너 생성

□ `docker run --name wordpress --link wordpressdb:mysql -p 80 -d wordpress`

```
[shjeon@localhost ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8e687cd78e24	wordpress	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:32773->80/tcp, :::32773->80/tcp	wordpress
83fadb541c22	mysql	"docker-entrypoint.s..."	3 minutes ago	Up 3 minutes	3306/tcp, 33060/tcp	wordpressdb

□ `--link` : 컨테이너간 접근 시 별명으로 접근 가능하도록 설정

■ `wordpressdb` 컨테이너를 `mysql` 별명으로 접근 가능

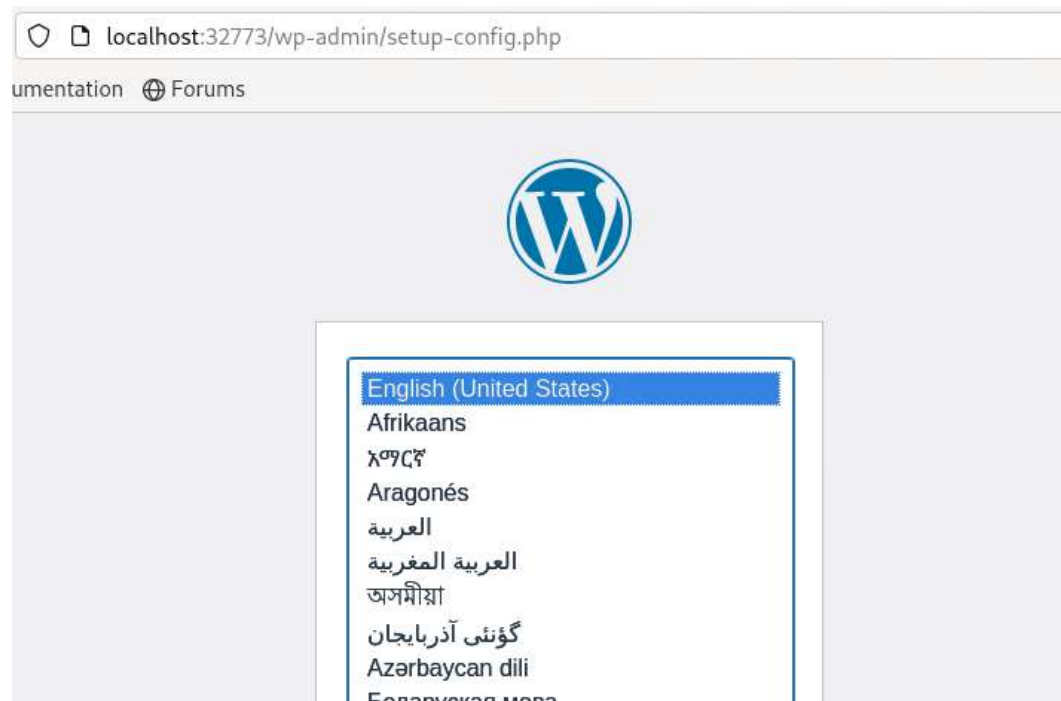
■ `--link`에 입력된 컨테이너가 중지 혹은 존재하지 않으면 실행 불가

□ `-p` : 호스트의 특정포트와 컨테이너의 **80**포트 연결

□ `wordpress` 웹 서버 **80**포트는 호스트의 **32773**으로 연결

컨테이너 애플리케이션 만들기

- 웹 서버 접속
 - localhost:3277



도커 이미지

■ 도커 이미지

- 도커 이미지는 도커 허브(Docker Hub)라는 중앙 이미지 저장소에서 다운로드함
- `docker create`, `docker run`, `docker pull` 명령어로 이미지 다운로드 가능
- 도커 계정 생성 후 이미지 업로드 및 다운로드 가능
- 도커 허브 비공개 저장소는 요금을 지불해야 사용 가능
- 이미지 저장소를 직접 구축해 사용 가능

도커 이미지

■ 이미지 검색하기

- docker search 이미지명
- docker search centos

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
centos	DEPRECATED; The official build of CentOS.	7620	[OK]	
kasmweb/centos-7-desktop	CentOS 7 desktop for Kasm Workspaces	39		
bitnami/centos-base-buildpack	Centos base compilation image	0		[OK]
couchbase/centos7-systemd	centos7-systemd images with additional debug...	8		[OK]
continuumio/centos5_gcc5_base		3		
datadog/centos-i386		0		
dokken/centos-7	CentOS 7 image for kitchen-dokken	5		
dokken/centos-8	CentOS 8 image for kitchen-dokken	3		
spack/centos7	CentOS 7 with Spack preinstalled	1		
dokken/centos-6	EOL: CentOS 6 image for kitchen-dokken	0		
atlas/centos7-atlasos	ATLAS CentOS 7 Software Development OS	0		

- STARS : 도커 사용자로부터 즐겨찾기 숫자
- OFFICAL : 공식 사이트에서 제공하는 이미지

도커 이미지

■ 이미지 검색하기

□ hub.docker.com

The screenshot shows the Docker Hub search results for the query 'centos'. The interface includes a search bar at the top with 'centos' entered, and navigation links for 'Explore', 'Pricing', 'Sign In', and 'Register'. On the left, there are filter sections for 'Products' (Images, Extensions, Plugins), 'Trusted Content' (Docker Official Image, Verified Publisher, Sponsored OSS), 'Operating Systems' (Linux, Windows), and 'Architectures' (ARM, ARM 64, IBM POWER, IBM Z, PowerPC 64 LE, x86, x86-64). The main content area displays three search results:

- centos** (DOCKER OFFICIAL IMAGE): Updated 8 months ago. DEPRECATED; The official build of CentOS. Linux: PowerPC 64 LE, ARM, 386, x86-64, ARM 64. Downloads: 1B+, Stars: 7.6K.
- kasmweb/centos-7-desktop** (VERIFIED PUBLISHER): By Kasm Technologies · Updated 4 days ago. CentOS 7 desktop for Kasm Workspaces. Linux: x86-64. Downloads: 1M+, Stars: 39.
- bitnami/centos-base-buildpack** (VERIFIED PUBLISHER): By VMware · Updated a year ago. Centos base compilation image. Linux: x86-64. Downloads: 50K+, Stars: 0. Pulls: 471 Last week. A line graph shows the pull trend over time.

도커 이미지

■ 이미지 생성하기

- 이미지 생성을 위한 컨테이너 생성
- `docker create -i -t --name image_test ubuntu:20.04`
- `docker commit [option] CONTAINER [REPOSITORY[:TAG]]`
- `docker commit -a "taba" -m "my first commit" image_test image_test:first`
- `-a` : author (이미지 작성자)
- `-m` : 커밋 메시지, 이미지 설명 입력

```
[shjeon@localhost ~]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
image_test	first	352645874148	About a minute ago	72.8MB
ubuntu	20.04	14be0685b768	5 weeks ago	72.8MB

도커 이미지

■ 이미지 히스토리 확인

□ docker inspect image_test:first

```
[shjeon@localhost ~]$ docker inspect image_test:first
[
  {
    "Id": "sha256:352645874148e743ca32bb6a57ce16959bf6410a0e61b79a639d3c1c25ea3aa9",
    "RepoTags": [
      "image_test:first"
    ],
    "RepoDigests": [],
    "Parent": "sha256:14be0685b7682b182af5b862c9638cb1cb4ca1a70bd5aa90deed96e9cca881e7",
    "Comment": "my first commit",
    "Created": "2023-08-02T16:27:05.833340973Z",
    "Container": "a8453059aa653d699d51d2ecd6988071d06698c1d85bc04ba7c572a3b912e453",
    "ContainerConfig": {
      "Hostname": "a8453059aa65",
```

□ docker inspect [이미지명]: 이미지의 상세 값 확인

도커 이미지

■ 이미지 정보 확인

□ docker history image_test:first

```
[shjeon@localhost ~]$ docker history image_test:first
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
352645874148	6 minutes ago	/bin/bash	0B	my first commit
14be0685b768	5 weeks ago	/bin/sh -c #(nop) CMD ["/bin/bash"]	0B	
<missing>	5 weeks ago	/bin/sh -c #(nop) ADD file:12f97b7b044d0d116...	72.8MB	
<missing>	5 weeks ago	/bin/sh -c #(nop) LABEL org.opencontainers...	0B	
<missing>	5 weeks ago	/bin/sh -c #(nop) LABEL org.opencontainers...	0B	
<missing>	5 weeks ago	/bin/sh -c #(nop) ARG LAUNCHPAD_BUILD_ARCH	0B	
<missing>	5 weeks ago	/bin/sh -c #(nop) ARG RELEASE	0B	

□ docker history [이미지명] : 이미지 생성 히스토리 출력

도커 이미지

■ 이미지 삭제

- `docker rmi image_test:first`
- 이미지를 사용 중인 컨테이너 존재, 이미지 레이어 삭제할 수 없음
- `docker rm -r` 로 삭제 가능

■ 이미지 추출

- Ubuntu20.04 이미지 추출
- `docker save -o new_ubuntu ubuntu:20.04`

```
[shjeon@localhost ~]$ docker save -o new_ubuntu.tar ubuntu:20.04
[shjeon@localhost ~]$ ls new_ubuntu.tar
new_ubuntu.tar
```

- `-o` 추출될 파일명 지정

■ 이미지 로드하기

- `docker load -i new_ubuntu`

도커 이미지

■ 이미지 배포

□ 파일배포

- 추출한 이미지 파일을 복사 후 저장
- 파일용량이 크고 도커엔진이 많을 때 배포가 어려움

□ 도커허브

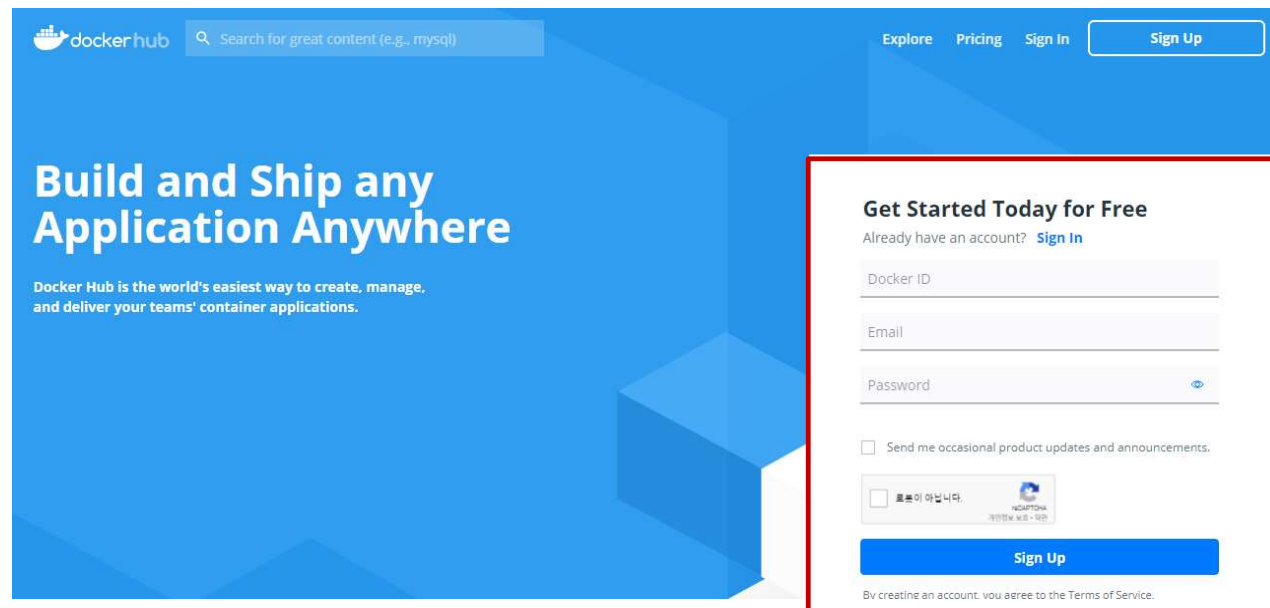
- 이미지 클라우드 저장소
- **Public** 무료저장소와 **Private** 유료 저장소 사용가능
- 무료 플랜에서는 공개 저장소는 무제한, 사설은 최대 1개 가능

□ 사설 레지스트리

- 사용자가 직접 도커 이미지 저장소를 직접 구성
- 저장소 서버, 저장공간을 사용자가 직접 관리해야함
- 회사 사내망 환경에서 이미지 배포 시 좋은 방법

도커 이미지

- 이미지를 공유하기 위해서는 Docker hub에 회원 가입 및 로그인
 - hub.docker.com



The screenshot shows the Docker Hub homepage with a blue background. The main heading is "Build and Ship any Application Anywhere". Below it, a subheading states: "Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications." On the right side, there is a "Get Started Today for Free" section enclosed in a red border. This section contains a "Sign In" link for existing users, input fields for "Docker ID", "Email", and "Password", a checkbox for "Send me occasional product updates and announcements.", a checkbox for "로그인 이 아닙니다." (I am not logged in) with a "PASSWORD" icon, and a blue "Sign Up" button. At the bottom of the sign-up box, it says "By creating an account, you agree to the Terms of Service."

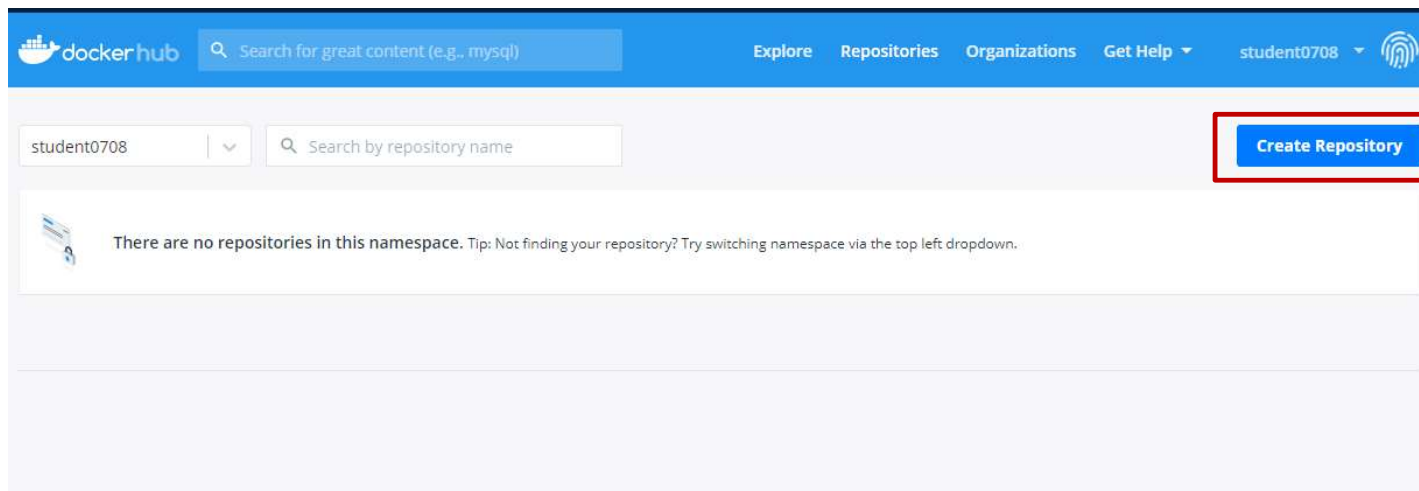
도커 이미지

- 로그인 후 상단 [Repositories] 메뉴를 클릭.



도커 이미지

- [Create Repository] 버튼을 클릭해 저장소 생성 화면으로 이동



도커 이미지

- 저장소의 이름을 지정
- 공개 혹은 개인으로 저장소를 생성

Create Repository

student0708

Description

Visibility

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

☒ **Public** Appears in Docker Hub search results

☐ **Private** Only visible to you

Build Settings (optional)

Autobuild triggers a new build with every **git push** to your source code repository. [Learn More](#)

Please re-link a GitHub or Bitbucket account

We've updated how Docker Hub connects to GitHub and Bitbucket. You'll need to re-link a GitHub or Bitbucket account to create new automated builds. [Learn More](#)



Disconnected



Disconnected

Cancel

Create

Create & Build

도커 이미지

■ 생성된 저장소를 확인

The screenshot shows the Docker Hub interface for a repository named 'student0708 / my_repository'. The repository is currently empty, with no tags or builds. The interface includes a search bar at the top, navigation tabs for 'General', 'Tags', 'Builds', 'Collaborators', 'Webhooks', and 'Settings'. A red box highlights the repository details section, which includes the repository name, a note that it has no description, and a 'Last pushed: never' status. To the right of the repository details, there are sections for 'Docker commands' (showing a push command) and 'Recent builds' (with a link to see build results). At the bottom, there is a 'Readme' section with a note that the repository description is empty.

dockerhub Search for great content (e.g., mysql) Explore Repositories Organizations Get Help student0708

Repositories student0708 / my_repository Using 0 of 1 private repositories. [Get more](#)

General Tags Builds Collaborators Webhooks Settings

Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts. [View preview](#)

student0708 / my_repository
This repository does not have a description
Last pushed: never

Tags and Scans **VULNERABILITY SCANNING - DISABLED**
This repository is empty. When it's not empty, you'll see a list of the most recent tags here. [Enable](#)

Docker commands [Public View](#)
To push a new tag to this repository.
`docker push student0708/my_repository:tagname`

Recent builds
[Link a source provider and run a build to see build results here.](#)

Readme
Repository description is empty. Click [here](#) to edit.

도커 이미지

- 다시 콘솔로 돌아가서 **login**을 한다.
- 로그인하기 위해서는 **[docker login]** 명령어를 사용한다. ID와 패스워드는 Docker허브에 가입한 계정 정보를 입력하면 된다.

```
[shjeon@localhost ~]$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to http
hub.docker.com to create one.
Username: shjeon0617
Password:
WARNING! Your password will be stored unencrypted in /home/shjeon/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

도커 이미지

- 저장소 이름과 동일하게 맞춰야 하므로 이미지를 새롭게 생성
 - `docker commit -a "taba" -m "my second commit" image_test ₩ shjeon0617/my_repository:first`

```
[shjeon@localhost ~]$ docker commit -a "taba" -m "my second commit" image_test shjeon0617/my_repository:first
sha256:1104405d938f832ca98dd74fdb96bb12cc60d93cd15b09f5cf18823aedb43f0f
[shjeon@localhost ~]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
shjeon0617/my_repository	first	1104405d938f	3 seconds ago	72.8MB
image_test	first	352645874148	33 minutes ago	72.8MB
ubuntu	20.04	14be0685b768	5 weeks ago	72.8MB

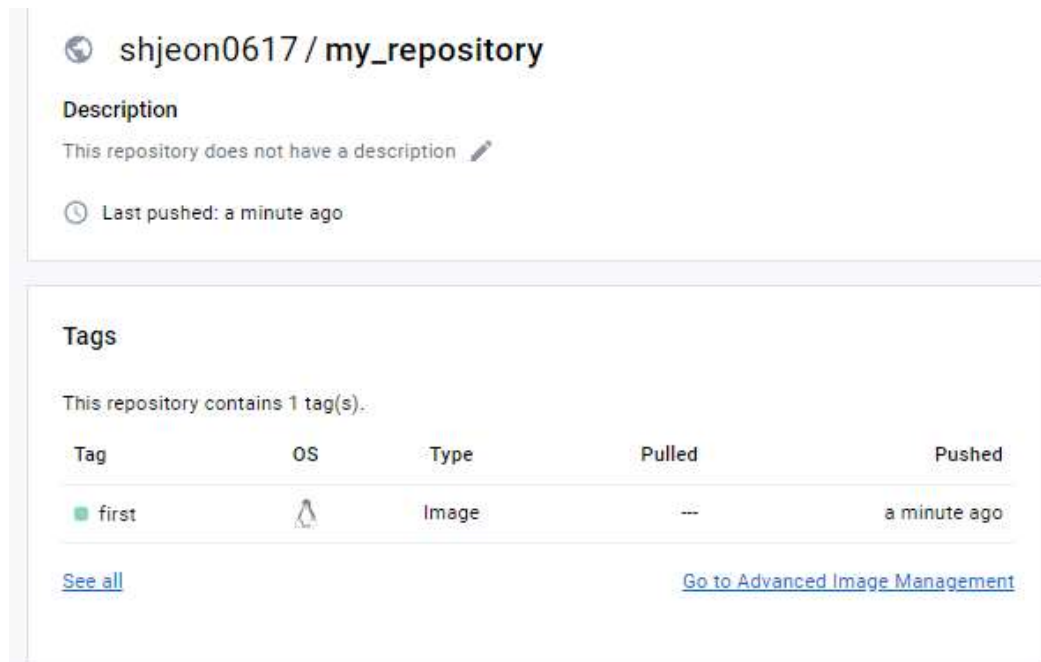
도커 이미지

- 이미지 업로드하기
 - `docker push shjeon0617/my_repository:first`
 - `docker push [option]` 이미지명:태그명

```
[shjeon@localhost ~]$ docker push shjeon0617/my_repository:first
The push refers to repository [docker.io/shjeon0617/my_repository]
f5bb4f853c84: Mounted from library/ubuntu
first: digest: sha256:7de2828f586fa3ce42bf9fe039cac9a204aca1335eb29107571da30a493f482d size: 529
```

도커 이미지

- 이미지 업로드하기
 - 업로드 후 docker hub에서 업로드 된 이미지 확인



The screenshot shows the Docker Hub interface for a repository named 'shjeon0617 / my_repository'. It includes a description field, a 'Last pushed' timestamp, and a 'Tags' section with a table of image tags.

shjeon0617 / my_repository

Description

This repository does not have a description

Last pushed: a minute ago

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
first	linux	Image	---	a minute ago

[See all](#) [Go to Advanced Image Management](#)