

Docker 사용해보기

목차

1. search 명령으로 이미지 검색하기
2. pull 명령으로 이미지 받기
3. images 명령으로 이미지 목록 출력하기
4. run 명령으로 컨테이너 생성하기
5. ps 명령으로 컨테이너 목록 확인하기
6. start 명령으로 컨테이너 시작하기
7. restart 명령으로 컨테이너에 접속하기
8. attach 명령으로 외부에서 컨테이너 안의 명령 실행하기
9. exec 명령으로 외부에서 컨테이너 안의 명령 실행하기
10. stop 명령으로 컨테이너 정지하기
11. rm 명령으로 컨테이너 삭제하기
12. rmi 명령으로 이미지 삭제하기

1. serach 명령으로 이미지 검색하기

Docker는 Docker Hub(<https://registry.hub.docker.com>)를 통해 이미지를 공유하는 생태계가 구축
유명 리눅스 배포판과 오픈 소스 프로젝트(Redis, Nginx 등)의 Docker 이미지는 모두 Docker Hub에서 구함
이미지와 관련된 모든 명령은 기본적으로 Docker Hub를 이용하도록 설정

docker search

```
$ sudo docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Official Ubuntu base image	383		
stackbrew/ubuntu	Official Ubuntu base image	40		
crashsystems/gitlab-docker	A trusted, regularly updated build of GitL...	19		[OK]
dockerfile/ubuntu	Trusted Ubuntu (http://www.ubuntu.com/) Bu...	15		[OK]
ubuntu-upstart	Upstart is an event-based replacement for ...	7		
cmfatih/phantomjs	PhantomJS [phantomjs 1.9.7, casperjs 1.1....	5		[OK]
dockerfile/ubuntu-desktop	Trusted Ubuntu Desktop (LXDE) (http://lxde...	5		[OK]
lukasz/docker-scala	Dockerfile for installing Scala 2.10.3, Ja...	5		[OK]
litaio/ruby	Ubuntu 14.04 with Ruby 2.1.2 compiled from...	5		[OK]

다양한 이미지가 검색될 것입니다.

보통 ubuntu, centos, redis 등 OS나 프로그램 이름을 가진 이미지가 공식 이미지입니다.

나머지는 사용자들이 만들어 공개한 이미지입니다.

2. pull 명령으로 이미지 받기

Docker Hub에서 우분투 리눅스 이미지를 받기

```
$ sudo docker pull ubuntu:latest
```

docker pull <이미지 이름>:<태그> 형식입니다. latest를 설정하면 최신 버전을 받습니다.

ubuntu:14.04, ubuntu:12.10 처럼 태그를 지정해 줄 수도 있습니다.

이미지 이름에서 pyrasis/ubuntu처럼 / 앞에 사용자명을 지정하면 Docker Hub에서 해당 사용자가 올린 이미지를 받습니다.
공식 이미지는 사용자명이 붙지 않습니다.

3. image 명령으로 이미지 목록 출력하기

받은 이미지의 목록을 출력

```
$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
ubuntu	latest	e54ca5efa2e9	Less than a second ago	276.1 MB

docker images 명령은 모든 이미지 목록을 출력합니다.

docker images ubuntu 처럼 이미지 이름을 설정하면 이름은 같지만 태그가 다른 이미지가 출력됩니다.

4. run 명령으로 컨테이너 생성하기

이미지를 컨테이너로 생성한 뒤 Bash 셸을 실행

```
$ sudo docker run -i -t --name hello ubuntu /bin/bash
```

docker run <옵션> <이미지 이름> <실행할 파일> 형식입니다.

여기서는 ubuntu 이미지를 컨테이너로 생성한 뒤 ubuntu 이미지 안의 /bin/bash를 실행합니다.

이미지 이름 대신 이미지 ID를 사용해도 됩니다.

- **-i**(interactive), **-t**(Pseudo-tty) 옵션을 사용하면 실행된 Bash 셸에 입력 및 출력을 할 수 있습니다.
- **--name** 옵션으로 컨테이너의 이름을 지정할 수 있습니다. 이름을 지정하지 않으면 Docker가 자동으로 이름을 생성하여 지정합니다.

5. ps 명령으로 컨테이너 목록 확인하기

다음 명령을 입력하여 모든 컨테이너 목록을 출력

```
$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6338ce52d07c	ubuntu:latest	/bin/bash	4 seconds ago	Exited (0)	Less than a second ago	hello

docker ps 형식입니다. **-a** 옵션을 사용하면 정지된 컨테이너까지 모두 출력하고, 옵션을 사용하지 않으면 실행되고 있는 컨테이너만 출력합니다.

6. start 명령으로 컨테이너 시작하기

정지한 컨테이너를 다시금 시작

```
$ sudo docker start hello
```

`docker start <컨테이너 이름>` 형식입니다. 컨테이너 이름 대신 컨테이너 ID를 사용해도 됩니다.

실행된 컨테이너 목록을 출력

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6338ce52d07c	ubuntu:latest	/bin/bash	15 minutes ago	Up 3 seconds		hello

7. restart 명령으로 컨테이너 재시작하기

OS 재부팅처럼 컨테이너 재시작

```
$ sudo docker restart hello
```

`docker restart` <컨테이너 이름> 형식입니다. 컨테이너 이름 대신 컨테이너 ID를 사용해도 됩니다.

8. attach 명령으로 컨테이너 접속하기

시작한 컨테이너에 접속. 명령을 실행한 뒤 엔터를 한번 더 입력하면 Bash 셸이 표시

```
$ sudo docker attach hello  
root@6338ce52d07c:/#
```

`docker attach <컨테이너 이름>` 형식입니다. 컨테이너 이름 대신 컨테이너 ID를 사용해도 됩니다.

- ✓ 우리는 /bin/bash를 실행했기 때문에 명령을 자유롭게 입력할 수 있지만, DB나 서버 애플리케이션을 실행하면 입력은 할 수 없고 출력만 보게 됩니다.
- ✓ Bash 셸에서 exit 또는 Ctrl+D를 입력하면 컨테이너가 정지됩니다. 여기서는 Ctrl+P, Ctrl+Q를 차례대로 입력하여 컨테이너를 정지하지 않고, 컨테이너에서 빠져나옵니다.

9. exec 명령으로 외부에서 컨테이너 안의 명령 실행하기

현재 컨테이너가 /bin/bash로 실행된 상태에서 /bin/bash를 통하지 않고 외부에서 컨테이너 안의 명령을 실행

```
$ sudo docker exec hello echo "Hello World"  
Hello World
```

`docker exec` <컨테이너 이름> <명령> <매개 변수> 형식입니다. 컨테이너 이름 대신 컨테이너 ID를 사용해도 됩니다. 컨테이너가 실행되고 있는 상태에서만 사용할 수 있으며 정지된 상태에서는 사용할 수 없습니다.

컨테이너 안의 `echo` 명령을 실행하고 매개 변수로 “Hello World”를 지정했기 때문에 Hello World가 출력됩니다.

`docker exec` 명령은 이미 실행된 컨테이너에 `apt-get`, `yum` 명령으로 패키지를 설치하거나, 각종 데몬을 실행할 때 활용할 수 있습니다.

10. stop 명령으로 컨테이너 정지하기

컨테이너 정지하기. 하기 전에 실행된 컨테이너 목록을 출력.

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6338ce52d07c	ubuntu:latest	/bin/bash	51 minutes ago	Up 2 minutes		hello

```
$ sudo docker stop hello
```

docker stop <컨테이너 이름> 형식입니다. 컨테이너 이름 대신 컨테이너 ID를 사용해도 됩니다.

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

✓ hello 컨테이너를 정지했기 때문에 아무것도 안 나옵니다.

11. rm 명령으로 컨테이너 삭제하기

생성된 컨테이너 삭제

```
$ sudo docker rm hello
```

`docker rm <컨테이너 이름>` 형식입니다. 컨테이너 이름 대신 컨테이너 ID를 사용해도 됩니다.

```
$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

✓ hello 컨테이너를 삭제했기 때문에 아무것도 안 나옵니다.

12. rmi 명령으로 이미지 삭제하기

이미지 삭제

```
$ sudo docker rmi ubuntu:latest
```

`docker rmi <이미지 이름>:<태그>` 형식입니다. 이미지 이름 대신 이미지 ID를 사용해도 됩니다.
`docker rmi ubuntu` 처럼 이미지 이름만 지정하면 태그는 다르지만 ubuntu 이름을 가진 모든 이미지가 삭제됩니다.

```
$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
------------	-----	----------	---------	--------------

✓ ubuntu 이미지를 삭제했기 때문에 아무것도 안 나옵니다.