

Docker 실습 2 - 기본 명령어

Docker 의 기본적인 명령

- 1) Docker pull
 - 2) Docker images
 - 3) Docker ps
 - 4) Docker run
 - 5) Docker exec
 - 6) Docker logs
 - 7) Docker stop
 - 8) Docker rm
 - 9) Docker rmi
-

Docker 의 기본적인 명령

1) Docker pull

- docker image repository 부터 Docker image 를 가져오는 커맨드입니다.

```
$ docker pull --help
```

- 예시)

```
$ docker pull ubuntu:20.04
```

- docker.io/library 라는 이름의 repository 에서 ubuntu:20.04 라는 image 를 여러분의 노트북에 다운로드 받게 됩니다.
(도커허브의 주소는 기본 생략)
- 참고사항
 - 추후 docker.io 나 public 한 docker hub 와 같은 repository 대신에, 특정 private 한 repository 에서 docker image 를 가져와야 하는 경우:
 - docker login 을 통해서 특정 repository 를 바라보도록 한 뒤, docker pull 을 수행하는 형태로 사용합니다.

2) Docker images

- 로컬에 존재하는 docker image 리스트를 출력하는 커맨드입니다.

```
$ docker images --help
```

- 예시)

```
$ docker images
```

3) Docker ps

- 현재 실행중인 도커 컨테이너 리스트를 출력하는 커맨드입니다.

```
$ docker ps --help
```

- 예시)

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

4) Docker run

- 도커 컨테이너를 실행시키는 커맨드입니다.

```
$ docker run --help
```

- 옵션에 대한 자세한 내용은 사용하게 될 때마다 하나씩 설명 예정
- 이미지가 없으면 pull 먼저 진행됨

- 예시)

```
$ docker run -it --name demo1 ubuntu:20.04 /bin/bash
```

- `-it` : `-i` 옵션 + `-t` 옵션

- container 를 실행시킴과 동시에 interactive 한 terminal 로 접속시켜주는 옵션
- `--name` : name
 - 컨테이너 id 대신, 구분하기 쉽도록 지정해주는 이름
- `/bin/bash`
 - 컨테이너를 실행시킴과 동시에 실행할 커맨드로, `/bin/bash` 는 bash 터미널을 사용하는 것을 의미합니다.

5) Docker exec

- Docker 컨테이너 내부에서 명령을 내리거나, 내부로 접속하는 커맨드

```
$ docker exec --help
```

- 예시)

```
$ docker run -it -d --name demo2 ubuntu:20.04
$ docker ps
```

- `-d` : 백그라운드에서 실행시켜서, 컨테이너에 접속 종료를 하더라도, 계속 실행 중이 되도록 하는 커맨드(daemon)

```
$ docker exec -it demo2 /bin/bash
```

- 아까와 동일하게 container 내부에 접속할 수 있는 것을 확인 가능

6) Docker logs

- 도커 컨테이너의 log 를 확인하는 커맨드

```
$ docker logs --help
```

- 예시)

```
$ docker run --name demo3 -d busybox sh -c "while true;
do $(echo date); sleep 1; done"
```

- test 라는 이름의 busybox 이미지를 백그라운드에서 도커 컨테이너로 실행하여, 1초에 한 번씩 현재 시간을 출력하는 커맨드

```
$ docker logs demo3
$ docker logs demo3 -f
```

- `-f` 옵션: 계속 watch 하며 출력

7) Docker stop

- 실행 중인 도커 컨테이너를 중단시키는 커맨드

```
$ docker stop --help
```

- 예시)

```
$ docker stop demo3
$ docker stop demo2
$ docker stop demo1
```

8) Docker rm

- 도커 컨테이너를 삭제하는 커맨드

```
$ docker rm --help
```

- 예시)

```
$ docker rm demo3
$ docker rm demo2
$ docker rm demo1
```

- `docker ps`, `docker ps -a` 에서 모두 출력되지 않는 것을 확인하실 수 있습니다.

9) Docker rmi

- 도커 이미지를 삭제하는 커맨드

```
$ docker rmi --help
```

- 예시)

```
$ docker images
# busybox, ubuntu 가 있는 것을 확인하실 수 있습니다.
$ docker rmi ubuntu:20.04
```