



1. 도커 - 설치부터 실행까지

1.1 도커 기본 명령어

1. run

- run 명령어를 사용하면 사용할 이미지가 저장되어 있는지 확인하고, 없으면 다운로드 (pull)한 후 컨테이너를 생성(create)하고 시작(start)함.
- 이때, 컨테이너는 정상적으로 실행됐지만 뭘 하라고 명령어를 전달하지 않았기 때문에 컨테이너는 생성되자마자 종료된다.

```
docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
```

<OPTIONS>

-d	detached mode (백그라운드 모드)
-p	호스트와 컨테이너의 포트를 연결
-v	호스트와 컨테이너의 디렉토리를 연결
-e	컨테이너 내에서 사용할 환경변수 설정
--name	컨테이너 이름 설정
--rm	프로세스 종료시 컨테이너 자동 제거
-it	-i와 -t를 동시에 사용한 것으로 터미널 입력을 위한 옵션
--network	네트워크 연결

<실습 1>

```
docker run --rm -it ubuntu:20.04 /bin/sh
```

- 컨테이너 내부에 들어가기 위해 sh를 실행하고 키보드 입력을 위해 -it 옵션을 줌.
- 프로세스가 종료되면 컨테이너가 자동으로 삭제되도록 --rm 옵션도 추가 (--rm 옵션이 없다면 컨테이너가 종료되더라도 삭제되지 않고 남아 있어 수동으로 삭제해야 함.)

1.2 컨테이너 명령어

명령어	설명	명령문
ps	실행 중인 컨테이너 목록 확인	docker ps docker ps -a # 중지된 컨테이너도 확인
stop	실행 중인 컨테이너를 중지	docker stop [OPTIONS] CONTAINER [CONTAINER...]
rm	종료된 컨테이너를 완전 제거	docker rm [OPTIONS] CONTAINER [CONTAINER...]
logs	컨테이너의 로그 확인 option : -f, -tail	docker logs [OPTIONS] CONTAINER

1.3 이미지 명령어

명령어	설명	명령문
images	도커가 다운로드한 이미지 목록을 확인	docker images [OPTIONS] [REPOSITORY[:TAG]]
pull	이미지 다운로드	docker pull [OPTIONS] NAME[:TAG @DIGEST]
rmi	이미지 삭제. 단, 컨테이너가 실행중일 때는 삭제되지 않음	docker rmi [OPTIONS] IMAGE [IMAGE...]

1.4 네트워크 명령어

명령어	설명	명령문
network creat	도커 컨테이너끼리 이름으로 통신할 수 있는 가상 네트워크 생성	docker network create [OPTIONS] NETWORK
network connect	기존에 생성된 컨테이너에 네트워크를 추가	docker network connect [OPTIONS] NETWORK CONTAINER