



## Ch2. 기본적인 도커 클라이언트 명령어 알아보기

🕒 생성일	@2022년 1월 19일 오후 8:52
☰ 태그	따라하며 배우는 도커와 CI환경

### 도커 이미지 내부 파일 구조 보기

`docker run [이미지 이름] ls`

- `docker` : 도커 클라이언트 언급
- `run` : 컨테이너 생성 및 실행
- 이미지 이름 : 이 컨테이너를 위한 이미지
- `ls` : 현재 디렉토리의 파일 리스트 출력
  - 이미지 이름 뒤에 다른 명령어를 더 붙여서 원래 이미지 안에 들어있는 기본 커맨드가 무시되고 `ls` 명령어가 실행

### 컨테이너들 나열하기

`docker ps` 로 컨테이너 정보 확인하기

- CONTAINER ID : 컨테이너의 고유한 아이디 해쉬값. (일부분만 표시)
- IMAGE : 컨테이너 생성 시 사용한 도커 이미지.
- COMMAND : 컨테이너 시작 시 실행될 명령어.
- CREATED : 컨테이너가 생성된 시간
- STATUS : 컨테이너의 상태 (Up, Exited, Pause)
- PORTS : 컨테이너가 개방한 포트와 호스트에 연결한 포트. (특별한 설정을 하지 않은 경우 출력x)
- NAMES : 컨테이너의 고유한 이름. 컨테이너 생성 시 설정하지 않으면 임의로 설정됨. 중복 불가능하고 `docker rename [original-name] [changed-name]` 명령어로 이름 변경 가능

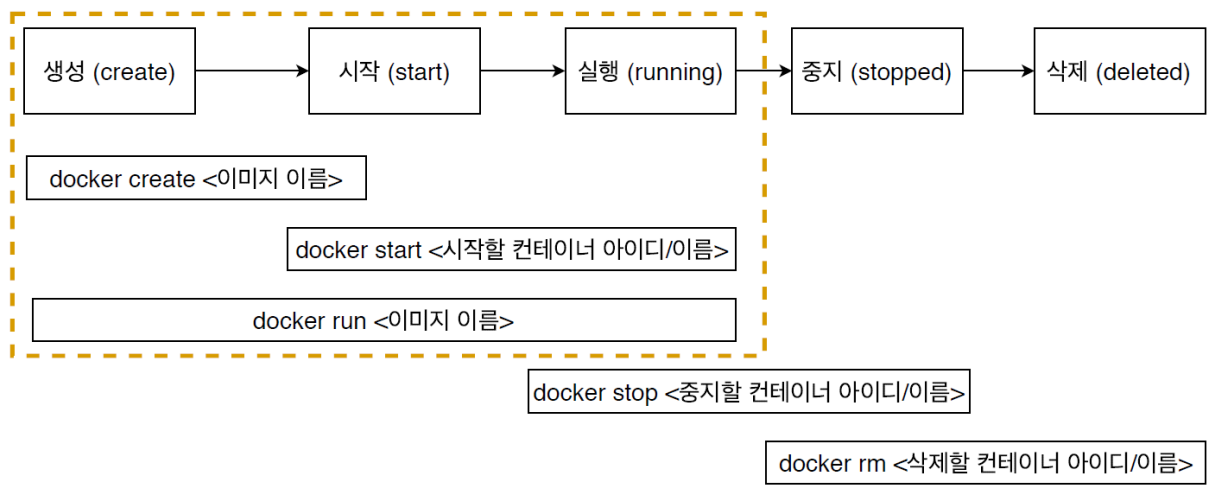
만약 도커 컨테이너의 원하는 항목만 보고 싶다면?

`docker ps --format 'table{{.Names}} \ table{{.Image}}'`

만약 모든 컨테이너를 나열하고 싶다면?

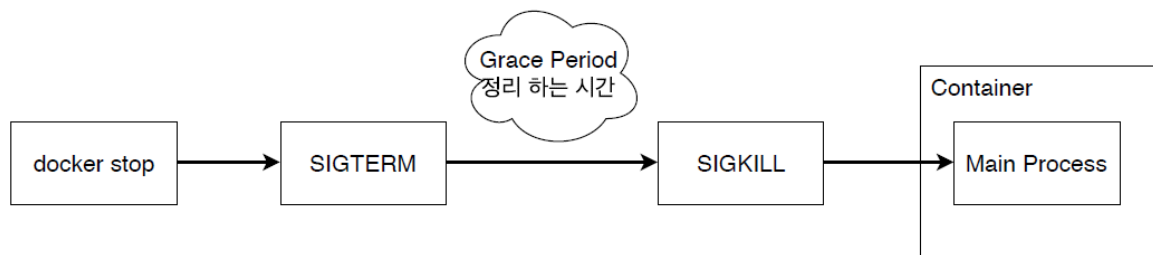
`docker ps -a`

### 도커 컨테이너의 생명 주기

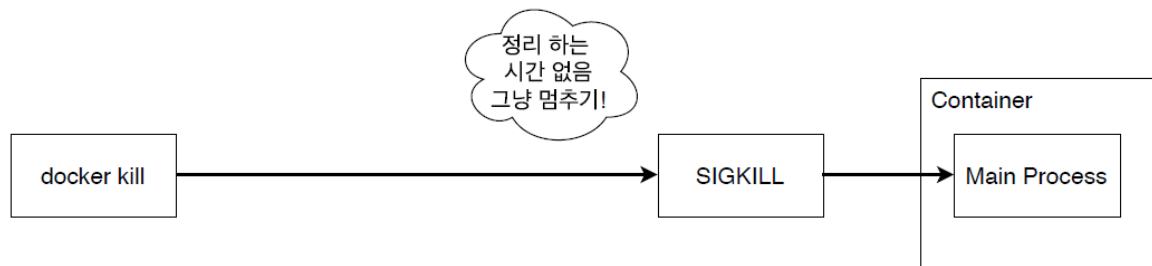


- `docker run` = `docker create` + `docker start`
  - `docker create` : 파일 스냅샷을 하드디스크에 넣어줌
  - `docker start` : 시작 시 실행될 명령어를 컨테이너에 넣어줌
    - `-a` 명령어 : attach / docker 컨테이너가 실행이 될 때 output을 화면에 표출해줌
- `docker stop` VS. `docker kill`
  - 컨테이너를 중지 하는 것은 동일하나 stop은 gracefully하게 중지하고 kill은 바로 중지

### ***docker stop <컨테이너 아이디/이름>***



### ***docker kill <컨테이너 아이디/이름>***



- `docker rm`
  - 실행중인 컨테이너는 먼저 중지한 후에 삭제 가능
  - 모든 컨테이너를 삭제하고 싶다면 `docker rm $(docker ps -a -q)`
  - 한번에 컨테이너, 이미지, 네트워크를 모두 삭제하고 싶다면 (도커 자체를 정리하고 싶다면) `docker system prune`
    - 실행중인 컨테이너에는 영향을 주지 않음!

## **실행중인 컨테이너에 명령어 전달**

이미 실행중인 컨테이너에 명령어를 전달하고 싶다면 `docker exec [컨테이너 아이디] [명령어]`

- `docker run` 은 새로운 컨테이너를 만들어서 실행
- `docker exec` 은 이미 실행중인 컨테이너에 명령어를 전달

## 레디스를 이용한 컨테이너 이해

- 레디스는 레디스 서버를 먼저 작동하고 나서 레디스 클라이언트를 실행한 뒤 명령어를 레디스 서버에 전달해야 함
- 레디스 클라이언트도 컨테이너 안에서 실행하여야 함.
  - `docker exec -it [컨테이너 아이디] redis-cli`
  - 이때 명령어에 `-it`를 붙여주어야 `redis-cli`에 머무를 수 있음.

## 실행 중인 컨테이너에서 터미널 생활 즐기기

- 지금까지 실행중인 컨테이너에 명령어를 전달하려면 `docker exec -it [컨테이너 아이디] [명령어]` 를 모두 입력하여야 함. → 매우 번거로움
- 따라서 `docker exec -it [컨테이너 아이디] sh` 로 컨테이너 안에 셸이나 터미널 환경으로 접근!!
  - `exec` 대신 `run`도 가능
  - `sh` 대신 `base-image`에 따라 다양하게 사용 가능 (`bash..` 등)
  - `shell`에서 빠져나오고 싶다면 `ctrl+D`