Docker 기본 명령어

도커 기본 명령어

- **컨테이너 목록 확인하기 (ps)**

  - docker ps [OPTIONS]

- **컨테이너 중지하기 (stop)**

  - docker stop [OPTIONS] CONTAINER [CONTAINER…]

- **컨테이너 제거하기 (rm) : 완전히 제거**

  - docker rm [OPTIONS] CONTAINER [CONTAINER…]

- **이미지 목록 확인하기 (images)**

  - 다운로드한 이미지 목록을 보는 명령어

  - docker images

- **이미지 다운로드하기 (pull)**

  - docker pull [OPTIONS] NAME[:TAG|@DIGEST]

- **이미지 삭제하기 (rmi)**

  - docker rmi [OPTIONS] IMAGE [IMAGE…]

컨테이너 둘러보기

- **컨테이너 로그 보기 (logs)**

  - docker logs [OPTIONS] CONTAINER

  로그?

  프로그램마다 로그 파일을 제각각 생기는데 어떻게 저렇게 로그가 나오는지하는 의문이 생긴다. 도커는 로그 파일을 자동으로 알아채는게 아니라 표준 스트림 중 stdout, stderr를 수집한다. 따라서 컨테이너에서 실행되는 프로그램의 로그 설정을 파일이 아닌 표준 출력으로 바꾸어야 한다. 단지 출력 방식만 바꾸는 것으로 모든 컨테이너는 로그에 대해 같은 방식으로 관리할 수 있게 된다.

  또 하나 중요한 점은 컨테이너의 로그 파일은 json 방식으로 어딘가에 저장이 된다. 로그가 많으면 은근히 파일이 차지하는 용량이 커지므로 주의해야 한다. 도커는 다양한 플러그인을 지원하여 json이 아닌 특정 로그 서비스에 스트림을 전달할 수 있다. 어느 정도 앱의 규모가 커지면 기본적인 방식 대신에 로그 서비스를 이용하는 것을 고려해야 한다.

- **컨테이너 명령어 실행하기 (exec)**

  - 컨테이너를 관리하다보면 실행중인 컨테이너에 들어가보거나 컨테이너의 파일을 실행하고 싶을

때가 있는데 SSH를 설치하는 것은 권장되지 않는다. **exec**라는 명령어를 사용하면 된다.

- docker exec [OPTIONS] CONTAINER COMMAND [ARG…]

  - run 명령어와 유사해 보이는데, 차이는 run은 새로 컨테이너를 만들어서 실행하고, exec는 실행중인 컨테이너에 명령어를 내리는 정도이다.

컨테이너 업데이트

- 컨테이너를 업데이트 하려면 새 버전의 이미지를 다운(**pull**)받고,

- 기존 컨테이너를 삭제(**stop, rm**)한 후,

- 새 컨테이너를 실행(**run**)하면 된다.

컨테이너를 삭제한다는 것은 컨테이너 안에 기존의 데이터가 사라진다는 것이다

삭제 시 유지해야 하는 데이터는 반드시 컨테이너가 아닌 외부 스토리지에 저장해야 한다.

가장 좋은 방법은

- AWS S3같은 클라우드 서비스 이용

- 데이터 볼륨(Data Volumes)를 컨테이너에 추가해서 사용

: 데이터 볼륨을 사용하면 해당 디렉토리는 컨테이너와 별도로 저장되고 컨테이너를 삭제해도 데이터가 지워지지 않는다.

- **mount**

Docker Compose

- 지금까지는 도커를 커맨드라인에서 명령어로 작업했으나, 컨테이너의 조합이 많아지고 여러 가지 설정이 추가되면 명령어가 금방 복잡해진다.

- 도커는 복잡한 설정을 쉽게 관리하기 위해 YAML 방식의 설정파일을 이용한 Docker Compose라는 툴을 제공한다.