

컨테이너

👤 생성자	양재 김
🕒 생성 일시	@2025년 4월 7일 오후 11:17
☰ 카테고리	Docker
🕒 최종 업데이트 시간	@2025년 4월 8일 오전 12:29

이미지(Image)

- 애플리케이션이 언제든지 실행 가능하도록 필요한 모든 요소를(소스코드, 구성요소, 종속성 etc.) 하나의 런타임 환경으로 패키징 한 정적 파일
- Docker 환경에서는 Dockerfile을 기반으로 build해 이미지 생성
- 불변성을 가지기 때문에 한번 생성되면 변경되지 않음

컨테이너(Container)

- 이미지를 실행한 격리된 논리적 공간(동작 중인 instance)
- Dockerfile을 build하여 생성된 이미지를 run 하여 동작하는 instance
- 애플리케이션과 종속 항목을 하나로 묶어서 실행하게 해주는 운영시스템을 가상화한 **경량의 격리된 프로세스**
- microVM
- 운영체제 수준의 가상화 제공(컨테이너의 OS는 커널이 존재X)
- Docker는 PAS(Platform As a Service)로 Host OS의 커널을 공유해서 컨테이너에서 사용
- 독립성을 가지며 다른 컨테이너에 영향을 주지 않는 **stateless** 환경

컨테이너 특징

- 다양한 환경에서 빠르고 효율적인 배포 가능
- 멀티 클러스터 환경으로 구축 가능하기 때문에 확장성 고려 가능
- 서버 구성, OS 설치, 네트워크 구성 등 반복적이고 불편한 작업에 시간 낭비를 줄이고 애플리케이션 개발 자체에 집중 가능

- 최소한의 Image를 통해 실행되기 때문에 경량 환경
 - 컨테이너에서 사용할 이미지 생성에서 가장 best practice는 이미지 경량화
- 언제나 프로세스 수준의 속도로 빠르게 실행 가능
- 한번에 여러 개의 컨테이너 동시 실행 가능(docker compose 사용)
- 어떤 환경에서도 동작 가능한 이식성
- 애플리케이션 환경에 대한 관리만 요구되기 때문에 지속적 서버 관리 비용 절감
- 개발과 운영 업무 분리로 세분화된 관리에 집중 가능 → DevOps workflow 구성에 최적화된 구조

컨테이너 타입

시스템 컨테이너 / OS 컨테이너

- Linux 이미지를 통해 배포되는 컨테이너
- 내부에 다양한 애플리케이션 및 라이브러리 도구 설치와 실행 가능
- 내부 PID 1번을 Host OS가 가짐

애플리케이션 컨테이너

- 독립적인 애플리케이션 실행을 위해 해당 서비스를 패키징하고 실행하도록 설계된 컨테이너
- 내부 PID 1번을 해당 애플리케이션이 가짐

Docker



- 여러 개층의 애플리케이션을 컨테이너로 분리하고 API로 실행하는 MSA 프로젝트에 유용
- 애플리케이션의 인프라(runtime)은 이미지를 통해 제공하고 public/private하게 공유 가능

- open share 방식으로 이미지를 docker hub에서 제공
- 제공된 이미지 기반으로 애플리케이션 서비스를 제공하고 이를 컨테이너화 (Containerization) 가능