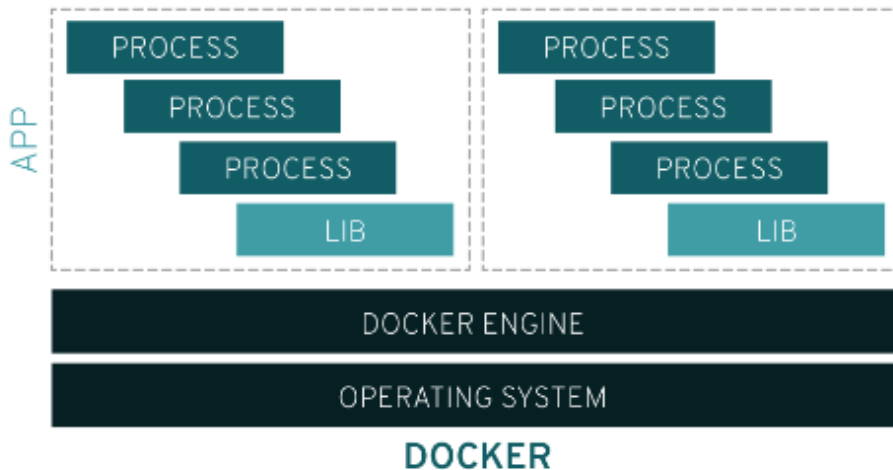


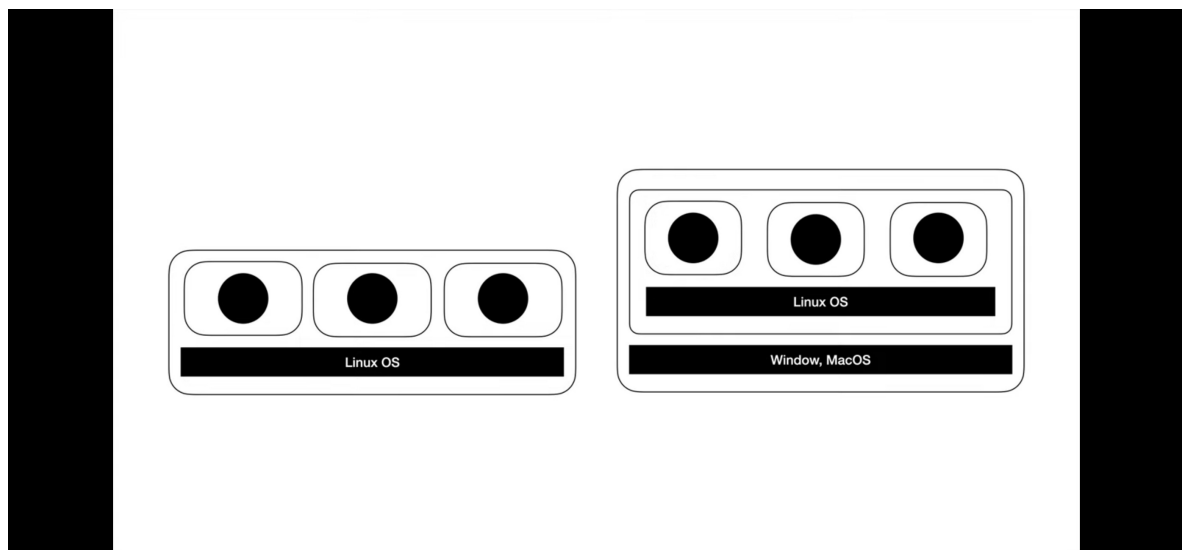
도커Docker

1. 개념

도커는 Linux 컨테이너를 만들고 사용할 수 있도록 하는 컨테이너화 기술



기본적으로 리눅스 운영체제 위에서 사용되기 위해 만들어졌으나, Window나 MacOS위에서도 "가상머신을 설치하고 Linux OS를 설치"해 사용 가능하다.



위의 과정을 도커가 알아서 해준다.

운영체제가 리눅스가 아니라면, 가상머신을 설치하기 때문에 어느 정도의 속도 저하는 있을 수 있다. 그럼에도 도커를 사용했을 때의 편의성 때문에 사용한다.

2. 도커 설치 방법

2-1. Hyper-V

Hyper-V는 x64 시스템을 위한 하이퍼바이저 기반의 가상화 시스템

Hyper-V는 Windows 10 Pro, Enterprise 및 Education 64비트 버전에서만 사용할 수 있고 Home 버전에서는 사용 할 수 없다(Hyper-V.bat 파일을 작성해서 사용할 수는 있음). 도커 데스크탑을 다운받아 보면 WSL 사용이 추천됨.

2-2. WSL

WSL은 Windows Subsystem for Linux의 약어로 윈도우 10에서 네이티브로 리눅스 실행 파일을 실행하기 위한 호환성 계층이다. 현재는 2번째 버전인 WSL2를 주로 사용한다.(도커 정식 지원도 WSL2부터)

WSL2를 사용하면 Hyper-V는 자동으로 사용된다.(Hyper-V를 통해 윈도우와 리눅스 간 파일 시스템의 내용을 공유한다.)

2-3. 도커 데스크탑 없이 사용

도커 데스크탑에는 도커 엔진, 도커 CLI클라이언트, 도커 빌드X, 도커 컴포즈, 도커 콘텐츠 트러스트, 쿠버네티스, 도커 자격 증명 도우미 등이 포함되어 있음. 하지만 도커 데스크탑의 유료화 및 사용 환경에 따라 도커 엔진만 필요한 경우, WSL2와 도커 엔진을 엮어 네이티브 도커처럼 사용할 수 도 있다.

***유료화 내용

- 중소기업(직원 수 250명 미만, 연간 매출 1,000만 달러 미만), 개인 용도, 교육 및 비상업적 오픈 소스 프로젝트에 무료 제공
- 위에 해당하지 않을 경우 전문적인 사용을 위해서는 유료 구독이 필요
- 정부 기관에도 유료 구독 필요
- Docker Pro, Team 및 Business 구독에는 Docker Desktop의 상업적 사용이 포함됩니다.

2-4. WSL2 및 도커 설치 방법

참조

<https://herojoon-dev.tistory.com/120>

https://goddaehee.tistory.com/313#google_vignette

***WSL 저장 위치

C드라이브에 저장할 경우 디스크 부족 문제가 야기될 수 있으므로 D드라이브 등 다른 곳에 저장

다른 드라이브로 옮길 경우 참조

<https://simryang.tistory.com/m/entry/wsl-wsl-d-%EB%93%9C%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%A1%9C-%EC%98%AE%EA%B8%B0%EA%B3%A0-%EA%B3%84%EC%A0%95-%EC%A0%95%EC%83%81%ED%99%94%ED%95%98%EA%B8%B0>

2-5. 도커는 기본적으로 리눅스 운영체제 위에서 실행되므로, WINDOW나 MAC OS 환경에서는 리눅스를 설치해야 한다.

WSL2를 설치하면서 우분투를 같이 설치한다.

리눅스에는 레드햇 계열(레드햇, CentOS 등)과 데비안 계열(데미안, 우분투, 페도라 등), 그리고 알파인 리눅스 등이 있는데 도커에서 공식적으로 데비안 계열 사용을 권장하고 있다.

3. 도커 컴포즈(Docker Compose)

도커 컴포즈란, 시스템 구축과 관련된 명령어를 하나의 텍스트 파일(Compose File)에 기재해 명령어 한 번에 시스템 전체를 실행하고 종료와 폐기까지 한 번에 하도록 도와주는 도구

YAML(YAML Ain't a markup Language) 포맷으로 기재한 정의 파일을 이용해 전체 시스템을 일괄 실행(run) 또는 종료 및 삭제(down) 할 수 있는 도구

도커 데스크톱 설치 시 함께 설치됨

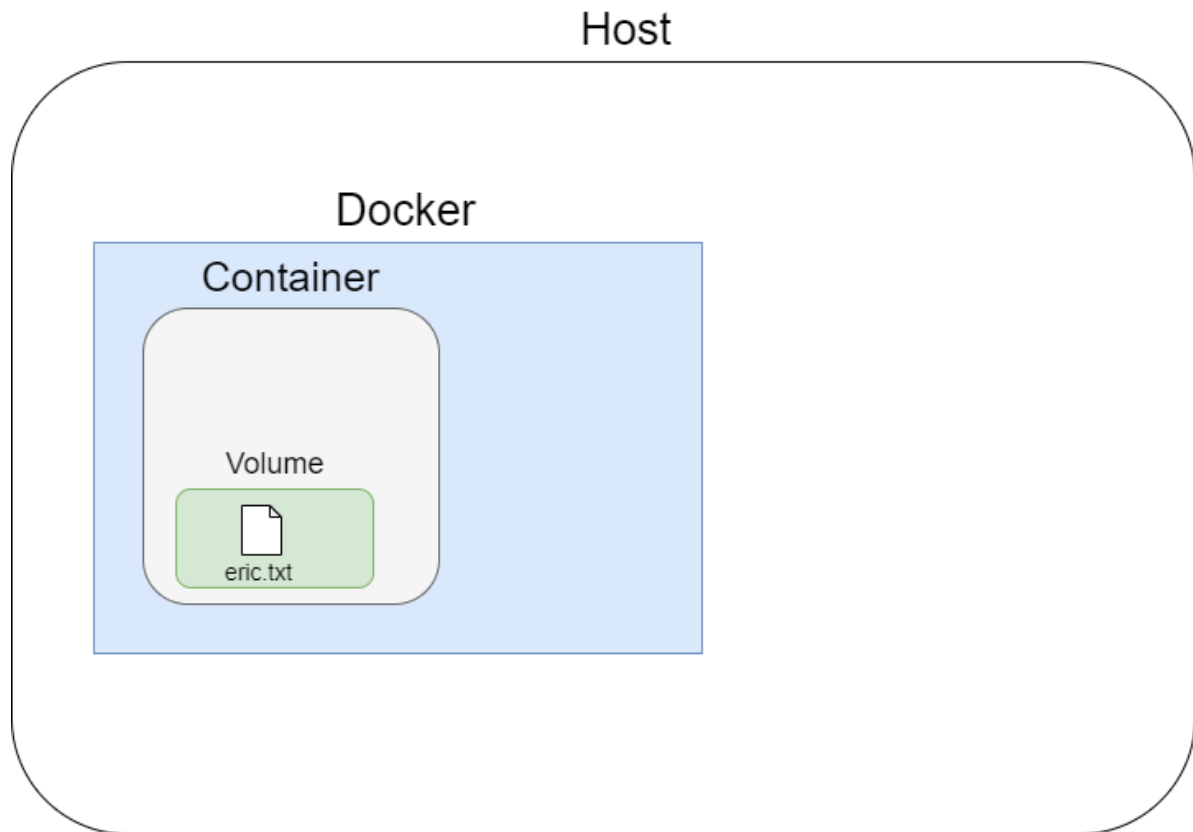
3-1. 도커 컴포즈 사용법 및 파일 작성법

참조

<https://devzzi.tistory.com/m/76>

4. 도커 볼륨

도커 컨테이너는 컨테이너마다 기본적으로 독립적인 저장소를 가지고 있다. 하나의 이미지로 여러 컨테이너를 만든다면 해당 컨테이너마다 독립적인 볼륨(하나의 파일 시스템을 갖춘 접근 가능한 저장공간)이 할당되고, 컨테이너가 삭제되면 해당 볼륨 또한 삭제된다. 즉 컨테이너가 삭제되면 내부에 저장된 데이터도 함께 삭제된다.

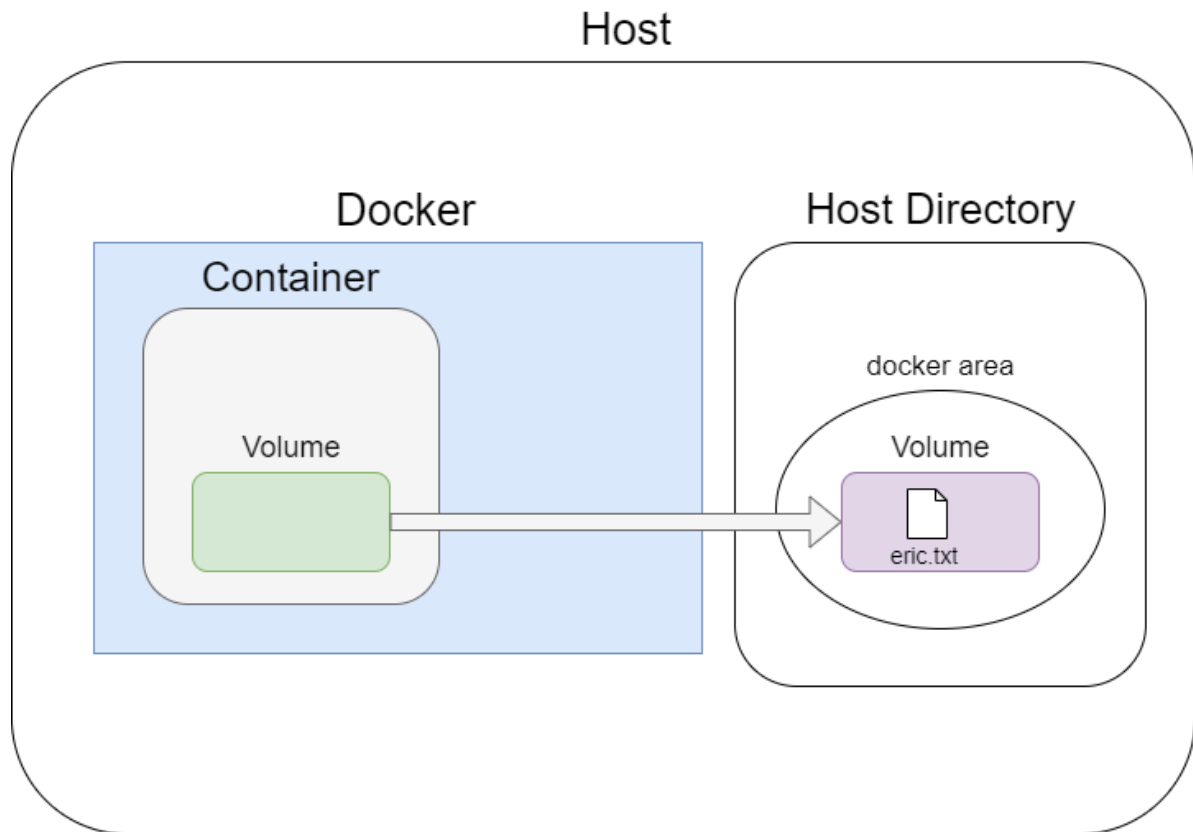


그래서 도커에서는 데이터의 영속성을 보장하기 위해 볼륨과 바인드 마운트를 이용한다. 볼륨과 바인드 마운트는 영속성을 보장하며 파일 시스템과 컨테이너를 분리해서 관리한다. 즉, 컨테이너를 지웠다가 다시 실행해도 도커 볼륨과 연결한 데이터는 그대로 유지된다.

4-1. 볼륨(Volume)

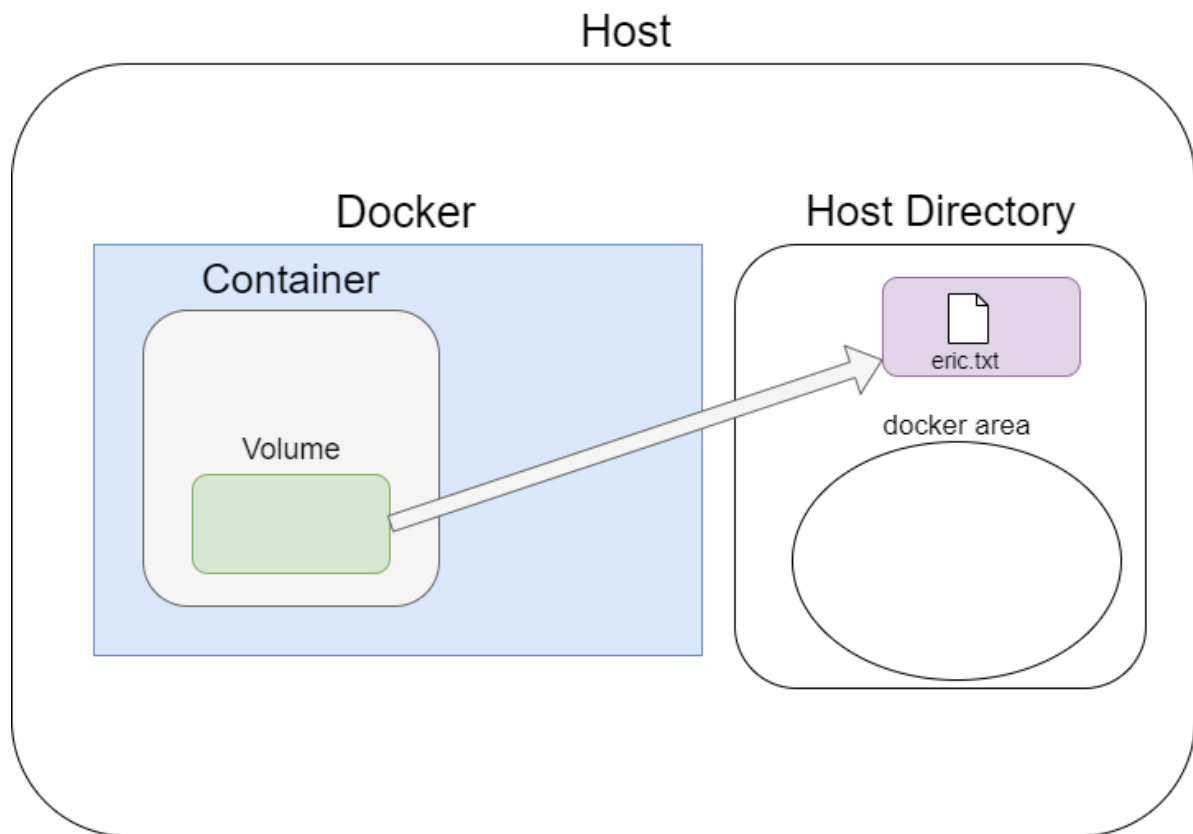
도커 공식 문서에서 권장하는 방식으로, 도커 엔진이 관리하는 도커 스토리지 디렉토리에 새 디렉토리를 생성하여 컨테이너 내부의 볼륨 데이터를 저장하는 방식
생성된 볼륨은 자동으로 호스트의 도커 스토리지 디렉토리인 `/var/lib/docker/volumes/~`에 저장된다.

볼륨을 컨테이너에 탑재하면 이 디렉토리가 컨테이너에 탑재되며, 도커에 의해 관리되고 호스트 시스템의 핵심 기능과 분리된다.



4-2. 바인드 마운트(Bind Mount)

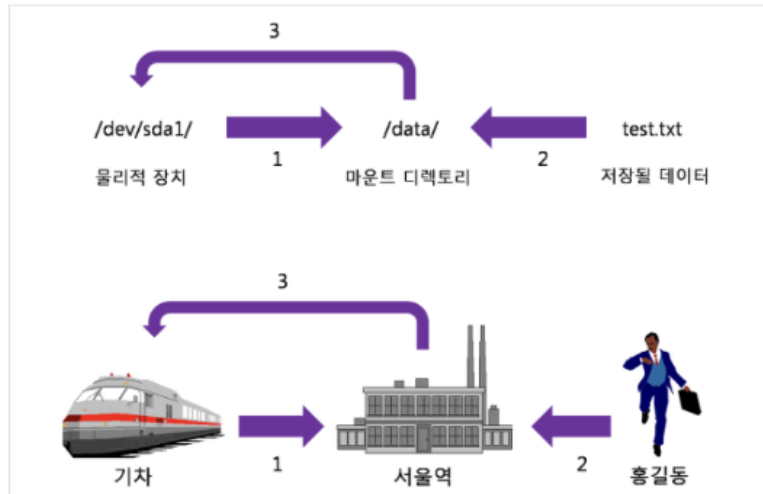
바인드 마운트는 도커가 관리하는 디렉토리가 아닌, 호스트 시스템의 파일이나 디렉토리가 컨테이너에 마운트되며 호스트 시스템의 절대 경로가 참조되는 방식. 도커 관리 없이 호스트 디렉토리와 마운트를 하다 보니 컨테이너에서 호스트의 파일 시스템에 접근하여 컨테이너에 지정된 파일이 아닌 다른 파일을 삭제하거나 수정할 수 있게 된다. 그래서 호스트 시스템의 비 Docker 프로세스에 영향을 줄 수 있고 보안에도 영향을 미칠 수 있으므로 도커 Volume을 사용하는 것을 권장한다.



*** 마운트(Mount)

물리적인 장치를 특정한 디렉토리 위치에 연결시켜 주는 과정

리눅스에서는 하드디스크의 파티션, CD/DVD, USB메모리 등을 사용하려면 특정한 위치에 연결을 해줘야 한다.



1. 기차는 서울역에서 정차한다. = 디스크를 디렉토리에 마운팅한다.
2. 홍길동은 서울역에서 기차를 탄다. = 데이터를 마운트 디렉토리에 저장한다.
3. 기차는 홍길동을 태우고 목적지까지 간다. = 실제 물리적으로 디스크에 저장된다.

여기에서 기차는 리눅스 상에 있는 물리적 장치이고, 서울역은 마운트된 디렉토리, 홍길동은 저장될 데이터다.
test.txt라는 데이터를 /data 아래에 저장하면 실제로 해당 데이터는 /dev/sda1 디스크에 저장이 된다.

참조

<https://www.44bits.io/ko/post/almost-perfect-development-environment-with-docker-and-docker-compose>