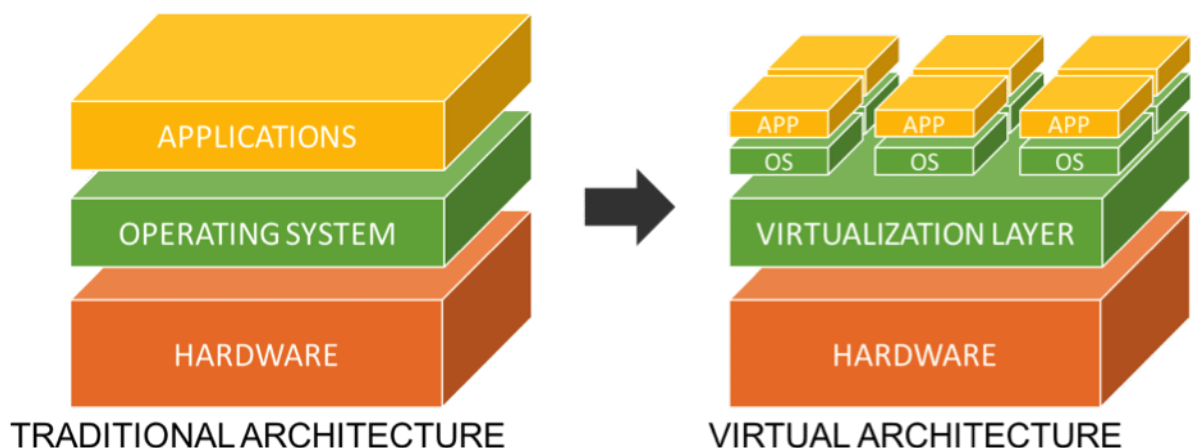


[OS] VM과 Docker 개요

≡ 태그

가상 머신(VM, Virtual Machine)

- 실재하는 컴퓨터 상 소프트웨어로 **논리적으로 만들어낸 컴퓨터**
- 하나의 물리 자원(컴퓨터) 위에 하나의 환경(OS)만 있는 것을 **효율화**하고자 **가상화**층을 만들고 그 위에 **OS를 새로 설치**하는 기법
- 하나의 컴퓨터를 **여러 명의 사용자가 동시에 사용할 수 있도록** 분할하거나, **운영 체제나 하드웨어 등의 구성을 달리하여 운영**하고자 할 때 주로 사용된다.
- 가상화 소프트웨어(ex - Vmware, VirtualBox)와 설치하려는 운영 체제의 ISO 파일(정보)이 필요하다.

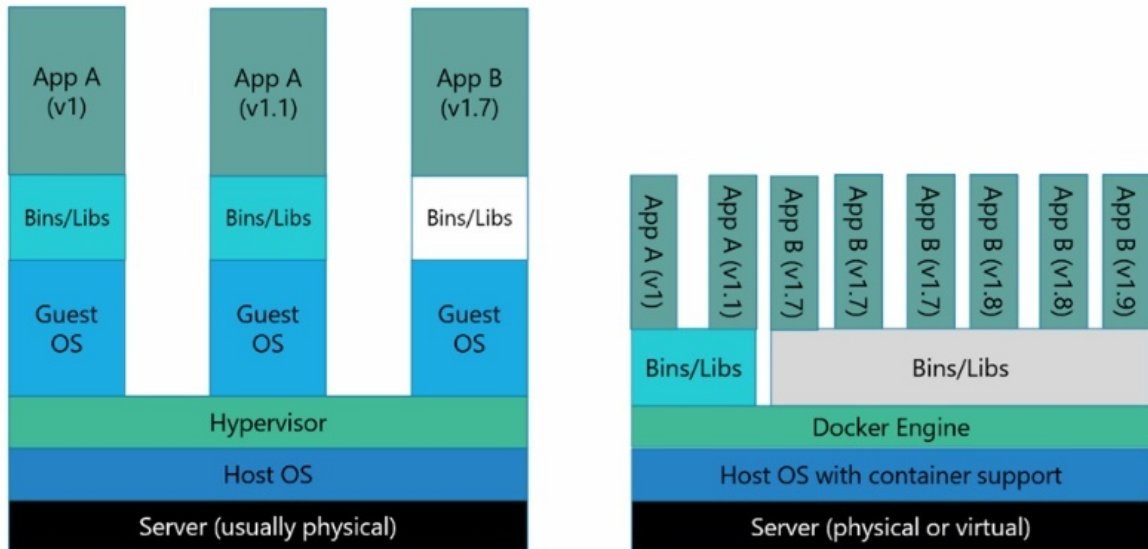


Docker

- Go 언어로 작성된 **리눅스 컨테이너 기반**으로 하는 **오픈소스 가상화 플랫폼**

컨테이너

- 가상화 기술 중 하나 (대표 : LXC(Linux Container))
- 기존 OS 가상화와 달리 **OS레벨의 가상화로 프로세스를 격리시켜 동작하는 방식**



좌 : VM, 우 : Docker

- 컨테이너 기반 가상화는 Docker 엔진 위에 Application 실행에 필요한 바이너리만 올라가게 된다.
- Host OS 그리고 Docker 엔진 위에서 바로 동작하며 Host의 커널을 공유.
- 커널 공유로 io 처리가 쉽게 되어 성능의 효율을 높일 수 있음.

Docker를 쓰는 이유

- 성능 향상, 뛰어난 이식성, 쉽게 Scale Out을 할 수 있는 유연성

Docker Image & Docker Hub

- 컨테이너를 실행할 수 있는 실행파일, 설정 값 등을 가지고 있는 것.
- Docker Image를 저장하고 배포하는 Docker Hub가 정말 잘 활성화 되어있음.

Docker File

- Docker hub에서 배포된 Image를 pull 하여 간단하게 컨테이너에 넣어 사용 가능
⇒ 배포판보다 보완하고 싶을 때 Docker File 사용!
- Docker File은 이미지 생성 출발점으로 이미지를 구성하기 위한 명령어들을 작성하여 이미지를 구성 가능. ⇒ Docker File을 읽을 수만 있다면 해당 이미지가 어떻게 구성되

어 있는지 알 수 있음

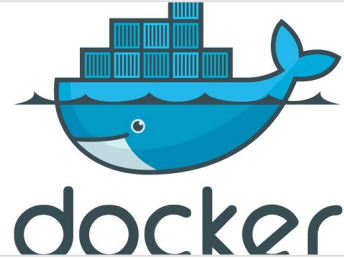
- [참고] Docker File 작성하여 Image 생성하기

추가 - Docker 설치부터 실행까지

[Docker] Docker 설치부터 실행까지 기본 사용법 정리

Docker 설치 \$ curl -fsSL https://get.docker.com/ | sudo sh\$
service docker start\$ docker version 설치 참... 쉽다.. docker 설치
후 시작한 다음 정상적으로 설치 되었나 확인을 해보면 됩니다.

☞ <https://khj93.tistory.com/entry/Docker-Docker-%EC%84%A4%EC%B9%98%EB%B6%80%ED%84%B0-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EC%8B%A4%ED%96%89%EA%B9%8C%EC%A7%80-%EA%B8%B0%EB%B3%B8-%EC%82%A4%EC%9A%A9%EB%B2%95>



참고

Virtual Machine(가상 머신)이란? (feat. 가상화 vs 클라우드)

가상 머신 (VM : Virtual Machine) 이란? - 실제하는 컴퓨터 상에 소프트웨어로 논리적으로 만들어낸 컴퓨터 - 하나의 물리 자원(컴퓨터) 위에 하나의 환경(OS)만 있는 것을 효율화하고자 가상화층을 만들고

☞ <https://joyfuls.tistory.com/48>



[Docker] Docker의 개념 및 핵심 설명

Docker란 Go언어로 작성된 리눅스 컨테이너 기반으로하는 오픈소스 가상화 플랫폼이다. 현재 Docker 0.9버전 부터는 직접 개발한 libcontainer 컨테이너를 사용하고 있다. 가상화를 사용하는 이유는?

☞ <https://khj93.tistory.com/entry/Docker-Docker-%EA%B0%9C%EB%85%90>

