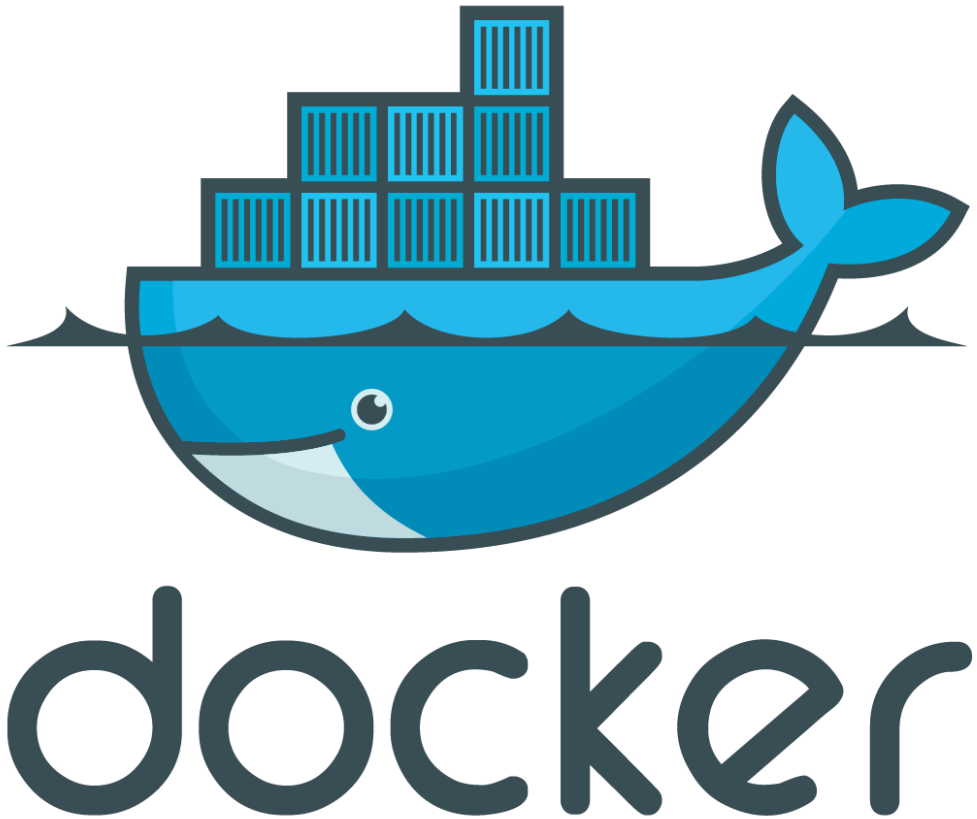
DOCKER

멀웨어 소모임-17정우혁

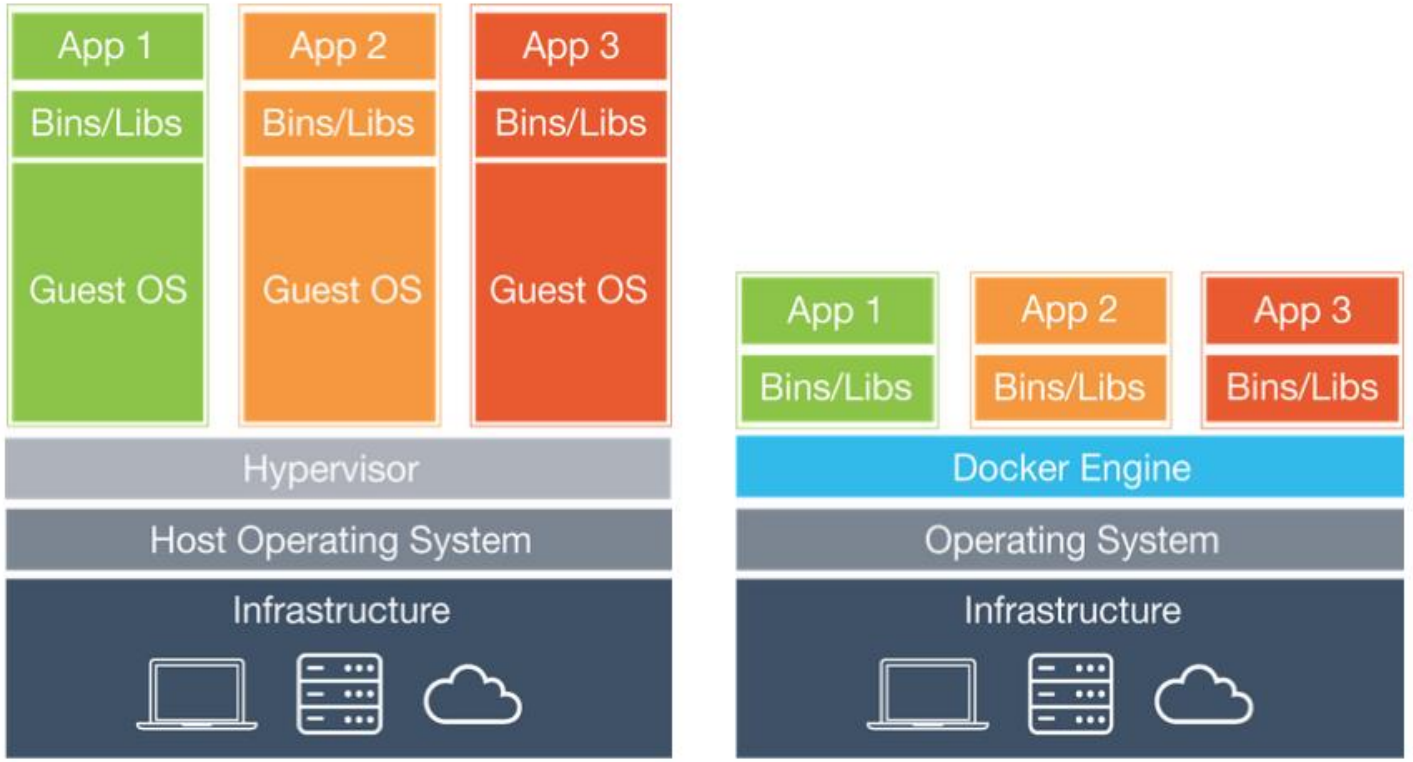
Docker 란?

- 컨테이너 기반의 오픈소스 가상화 플렛폼.
- Azure, Google Cloud 등을 docker 위에서 사용할 수 있다.



Docker vs VM

- 가상화:컴퓨터 리소스 추상화?
- Docker 가 Overhead 가 적다!



Docker 장/단점

- 장점: 1. 쉬운 유지보수 (union mount, layer 개념)
  2. sharing (docker hub)
  3. 가벼움&minimal overhead.(no OS!)

- 단점: kernel의존적(only in Linux) ->
  (Window, Mac은 docker 내부의 가상머신이 또 돌아간다...)
  -> window, mac 회사 엔지니어들이 해결을 위해 투입되었다.

Docker 구성

- 1. Docker registries -> github! Sharing 기능
- 2. Docker Image
- 3. Docker engine
- 4. Docker container

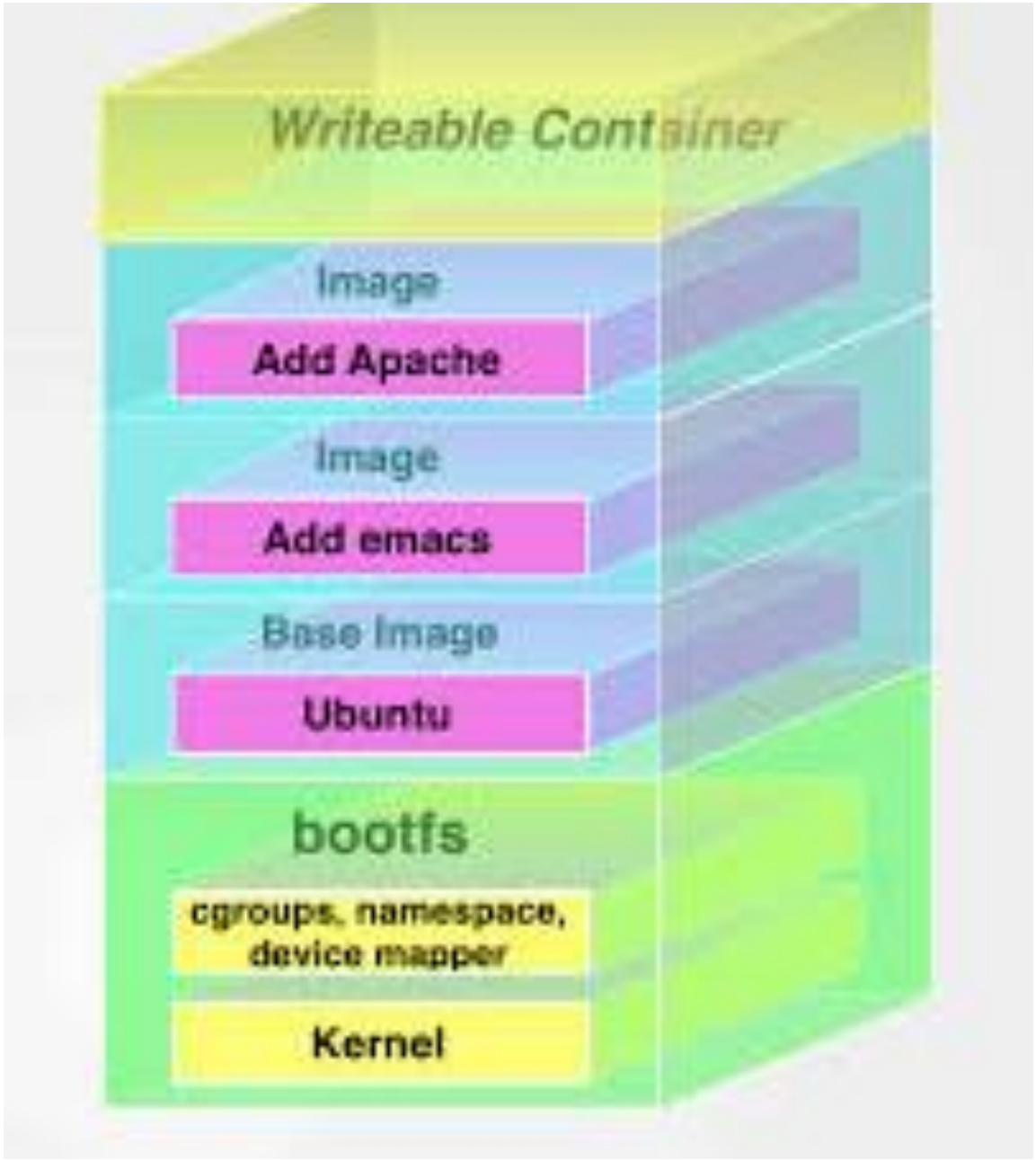
Docker registry

- https://hub.docker.com/ <- 파일 업로드 가능

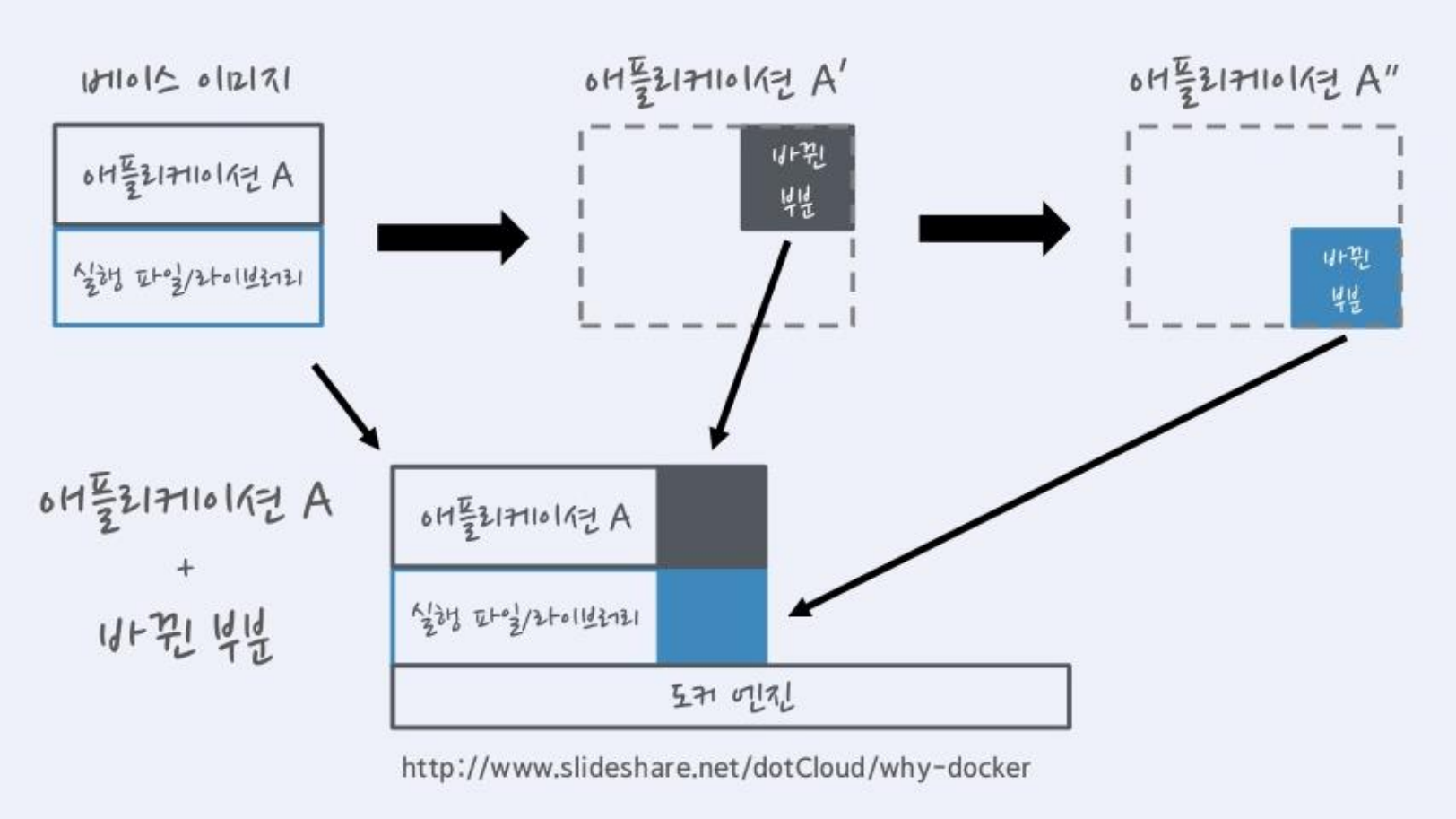
- 또는 스스로 registry를 구성해서 활용 가능.

Docker Image

- 쉬운 유지 보수
- Read-only
- Writeable Container
- Union mount -> Layer 개념



Docker 유지보수


베이스 이미지
어플리케이션 A
실행 파일/라이브러리
어플리케이션 A'
바뀐 부분
어플리케이션 A"
바뀐 부분
어플리케이션 A
+
바뀐 부분
어플리케이션 A
실행 파일/라이브러리
도커 엔진
http://www.slideshare.net/dotCloud/why-docker

OCI(Open Container Initiative)-Chroot

- Chroot:프로세스의 root directory 를 변경
- -> 프로세스가 엑세스 할 수 있는 디렉토리의 제한

- 하지만, chroot 만으로는 네트워크&프로세스 제한이 불가, 리소스 제어 불가.

OCI– Cgroups

- Cgroups: 리소스에 대한 제어를 가능하게 해주는 리눅스 커널의 기능.
- 제어 가능한 리소스: 파일 시스템, 프로세스, 메모리, CPU, I/O, 네트워크
- -> 컨테이너 별로 리소스를 할당 및 제한이 가능하다.

OCI- namespace

- Namespace: 하나의 system에서 수행 되나, 각각 별개의 공간인 것처럼 격리된 환경을 제공하는 lightweight 가상화 기술.
- Clone() syscall을 써서 완전히 독립된 프로세스(컨테이너)를 만든다.
- hostname, network, PID, user 등을 독립 시킬 수 있다.

OCI

- “Jail”의 탄생:
  chroot와 cgroups를 적절히 활용하여, 부모로 부터 독립되고,
  제한된 리소스를 제공받는 곳(즉, 컨테이너!)을 생성가능.
- -> Linux LXC

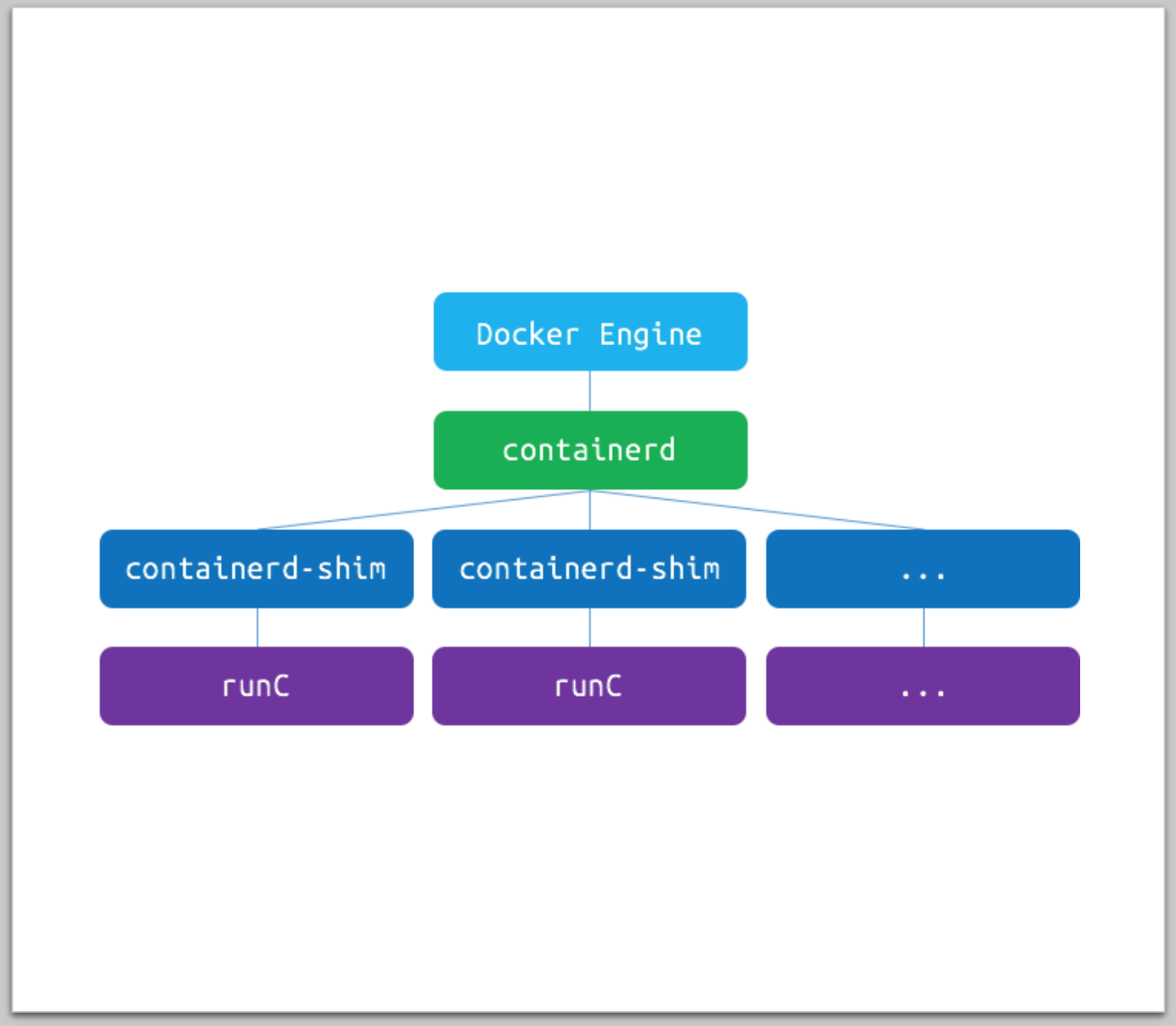
- Docker runC는 Linux LXC를 바탕으로, Jail 기능에 서버 관리 등 Docker 목적에 맞는 기능들을 추가로 넣어둔 OCI 이다.

Docker Container

- runC로 만들어진 완전히 독립된 공간!
- runC에 의해 할당된 컨테이너 별 분할 자원:
- 프로세스 테이블: 컨테이너 각각이 테이블을 가짐 -> 다른 프로세스 엿볼 수 없다.(컨테이너간 독립성 보장)
- 파일 시스템: chroot 이용해서 분리 + namespace
- 네트워크: 독립된 ip 가질 수 있다. network namespace 이용.
- CPU, 메모리장치 같은 리소스: cgroup. 컨테이너 별 리소스 접근 범위 제한.

Docker Engine

- Docker Engine: 이미지, 네트워크, 디스크 관리.
- Containerd: OCI 구현체(runC)를 이용해 컨테이너를 관리해주는 데몬.
- Engine과 각 containerd의 완전한 분리-> Engine update 한 뒤 재부팅 할 때 runC들을 종료 안 해도 된다.



Docker Security

- 사람들이 올리는 Docker Image를 막 받아쓰면 안된다!
- CVE-2014-9357 : 조작된 이미지를 써서, host의 root 권한 탈취 취약점
- CVE-2016-8867: 조작된 이미지가 docker engine내 취약점 이용, user permission 을 무시하고 컨테이너에 무단 침입 가능

- 사용하는 리눅스의 커널 취약점 -> host 포함 모든 컨테이너 취약점

끝내며

1.오 개념이 있을 수 있습니다! 혹시 틀린 게 있다면 고쳐주세요

2. 질문에 충분히 대답을 못할 수 있습니다! 구해놓은 자료를 업로드 하겠습니다... 죄송해요 ㅜㅜ

어려웠다..!