

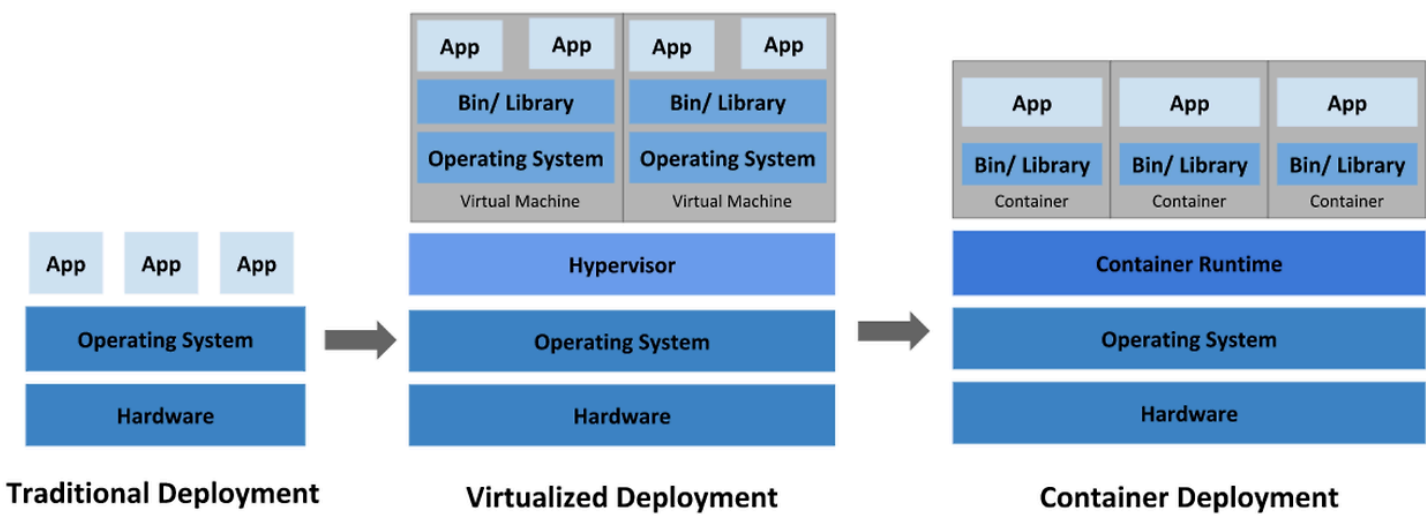
# 컨테이너(container)가 뭐예요?

대충말하면 컨테이너는 애플리케이션을 환경에 구매 받지 않고 실행하는 기술입니다.  
정확하게 말하자면 운영체제에서 실행되는 프로세스를 격리(Isolation)하여 별도의 실행 환경을 제공해주며, 해당 프로세스는 운영체제 상에서 실행되는 유일한 프로세스인 것처럼 작동하는 기술입니다.  
즉, 운영체제에서 실행되는 여러 프로세스는 컨테이너라는 개념으로 격리되어 **별도의 운영 환경을 제공해주는 기술**입니다.

## 왜 컨테이너라고 부르죠?



어떤 물체를 격리하는 공간의 뜻을 가지고 있는 컨테이너는 가로, 세로, 높이의 크기가 세계적으로 표준화되어 있습니다. 컨테이너의 크기가 통일되어 있기 때문에 화물 트레일러가 옮기고 그대로 대형 선박에 컨테이너를 옮겨 적재 할 수 있습니다. 또한, 컨테이너 표준을 지키면 물류에 대한 계약, 보관, 운송, 보험 등에 대해서 동일한 기준으로 비즈니스 할 수 있습니다.



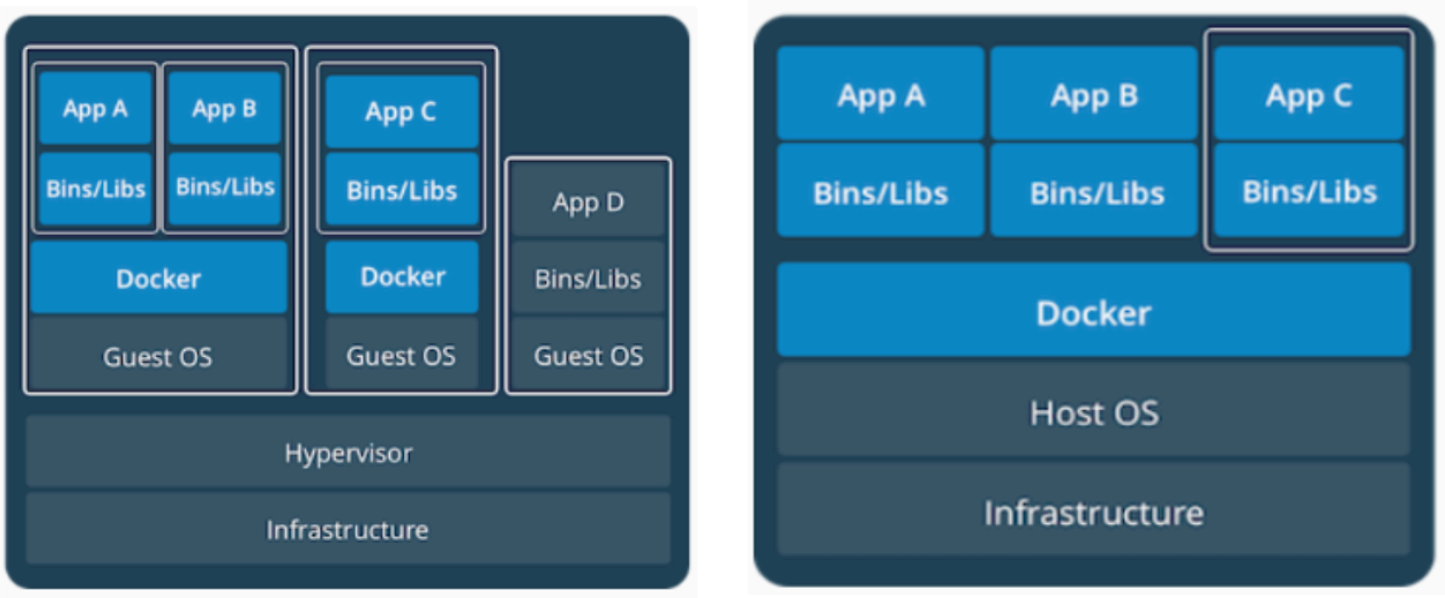
리눅스 기술을 사용하여 선박의 컨테이너처럼 프로세스가 사용하는 자원을 격리하며 가상머신과 마찬가지로 애플리케이션 관련 라이브러리 및 종속 항목과 함께 패키지로 묶어 소프트웨어 서비스 구동을 위한 격리 환경을 마련해줍니다.

## 컨테이너의 장점

- 가벼움
- 탄력성
- 유지 관리 효율

각각의 장점들에 대해서 한가지씩 보도록 하겠습니다.

가벼움



사용자의 Request Traffic 이 증가함에 따라, 가상머신이나 컨테이너를 추가적으로 배포할 때

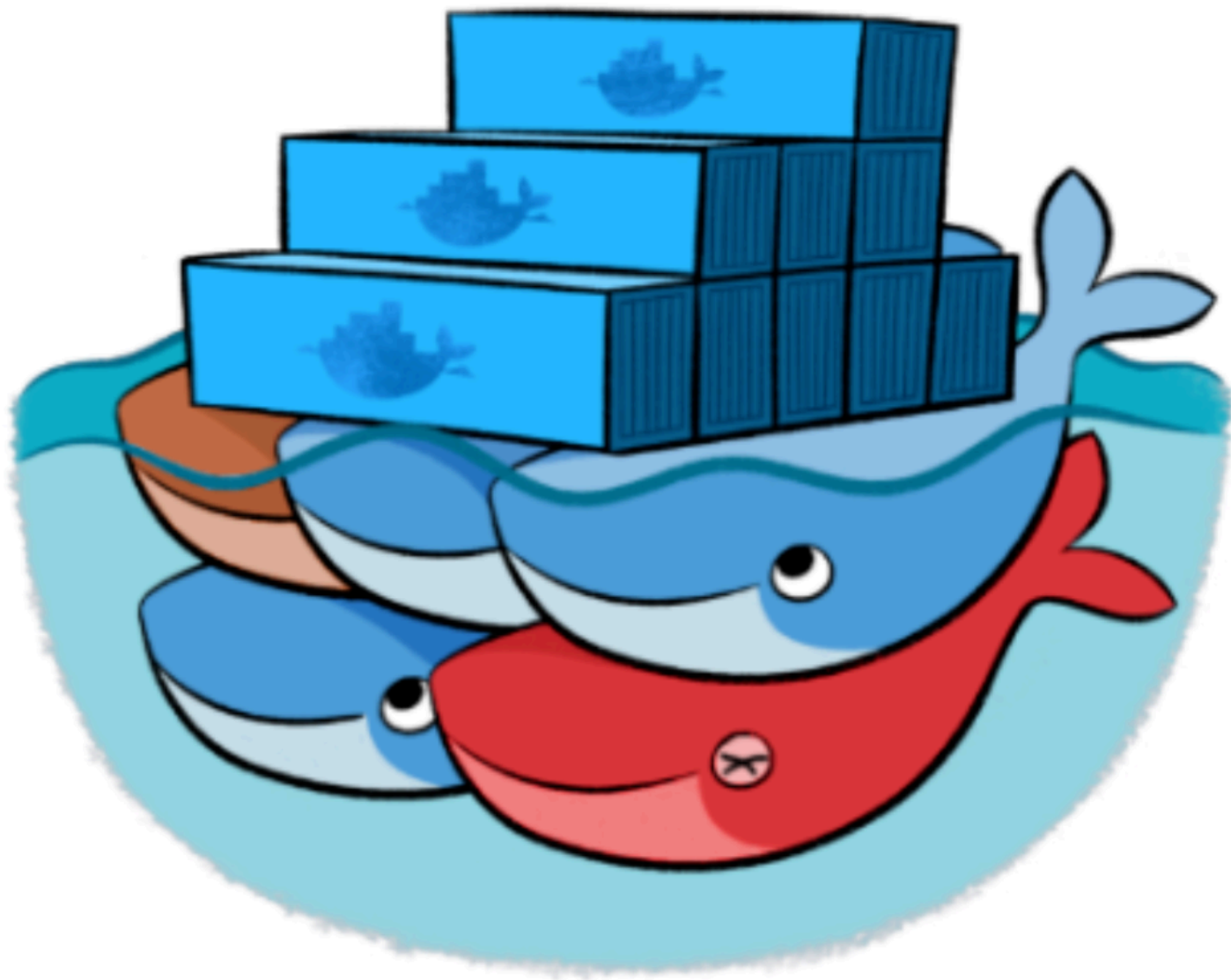
가상머신의 크기는 최소 GB 단위 이지만, 컨테이너의 경우 Guest OS가 없기에 MB 단위의 크기를 가집니다.

탄력성



컨테이너는 Linux, Windows, 가상머신, Data Center, Public Cloud 등 어느 환경에서나 구동이 되므로 개발 및 배포가 크게 쉬워집니다.

유지관리 효율



운영 체제 커널이 하나밖에 없기 때문에 운영 체제 수준에서 업데이트 또는 패치 작업을 한 번만 수행하면 변경 사항이 모든 컨테이너에 적용됩니다.

이를 통해 서버를 더 효율적으로 운영하고 유지 관리할 수 있습니다.

## 컨테이너 아키텍처

리눅스 시스템에서 컨테이너를 이용하여 격리 구조를 만드는 기법은 격리를 담당하는 Linux Namespace와 리소스를 제어하는 Control Group(cgroup)을 사용하여 격리된 컨테이너 환경을 제공합니다.

리눅스 시스템에서 네임스페이스는 기본적으로 단일 네임스페이스를 사용하여 동작하며, 네임스페이스는 다음과 같은 종류로 있습니다.

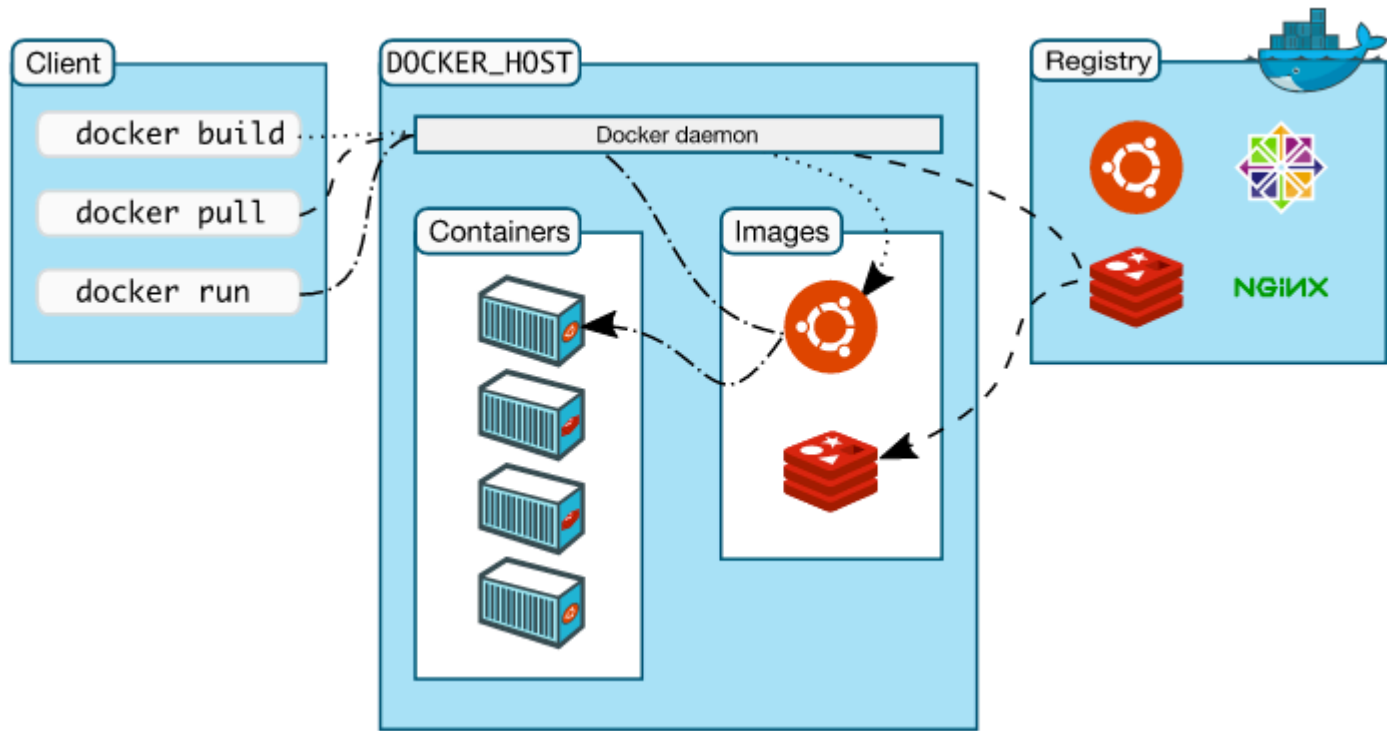
- 마운트 포인트
- 프로세스
- 네트워크 - IPC
- UTS
- 사용자

제어 그룹은 프로세스 또는 컨테이너가 사용할 수 있는 리소스의 양을 제한할 수 있습니다. 제어 그룹이 제한할 수 있는 리소스를 다음과 같습니다.

- CPU
- 메모리
- 네트워크 대역폭
- 디스크 입출력

# 도커 컨테이너

컨테이너 기술은 오래전부터 존재하던 기술이었지만 Docker Inc가 만든 도커 플랫폼의 등장으로 컨테이너 플랫폼에 대해 널리 알려지고 쉽게 사용할 수 있게 되었습니다. 도커는 namespace와 제어 그룹 기술을 사용하여 애플리케이션을 패키징, 배포, 실행하는 플랫폼 입니다.



## 이미지

- 실행할 애플리케이션과 라이브러리 및 환경을 하나의 패키지로 묶은 것

## 컨테이너

- 이미지를 실행한 컨테이너

## 레지스트리

- 이미지를 저장하고 공유할 수 있는 스토리지
- 도커 허브 등과 같은 공용 레지스트리와 개인 및 특정 시스템만 사용할 수 있는 사설 레지스트리를 직접 구축 가능

도커는 컨테이너를 주류 기술로 만든 최초의 컨테이너 플랫폼으로, 쿠버네티스에서 기본적으로 애플리케이션을 구동하기 위한 구성요소 중 하나입니다. 그러나 컨테이너를 실행하기 위한 플랫폼은 도커만 있는 것은 아닙니다. 최초의 쿠버네티스에서는 도커만 컨테이너 플랫폼으로 지원했지만, 지금은 컨테이너 형식 및 컨테이너 런타임에 대한 개방된 표준을 만드는 OCI(Open Container Initiative)가 탄생했고 쿠버네티스는 OCI 표준의 컨테이너 대부분을 지원합니다. 대표적으로 도커의 대안인 rkt 가 있습니다.

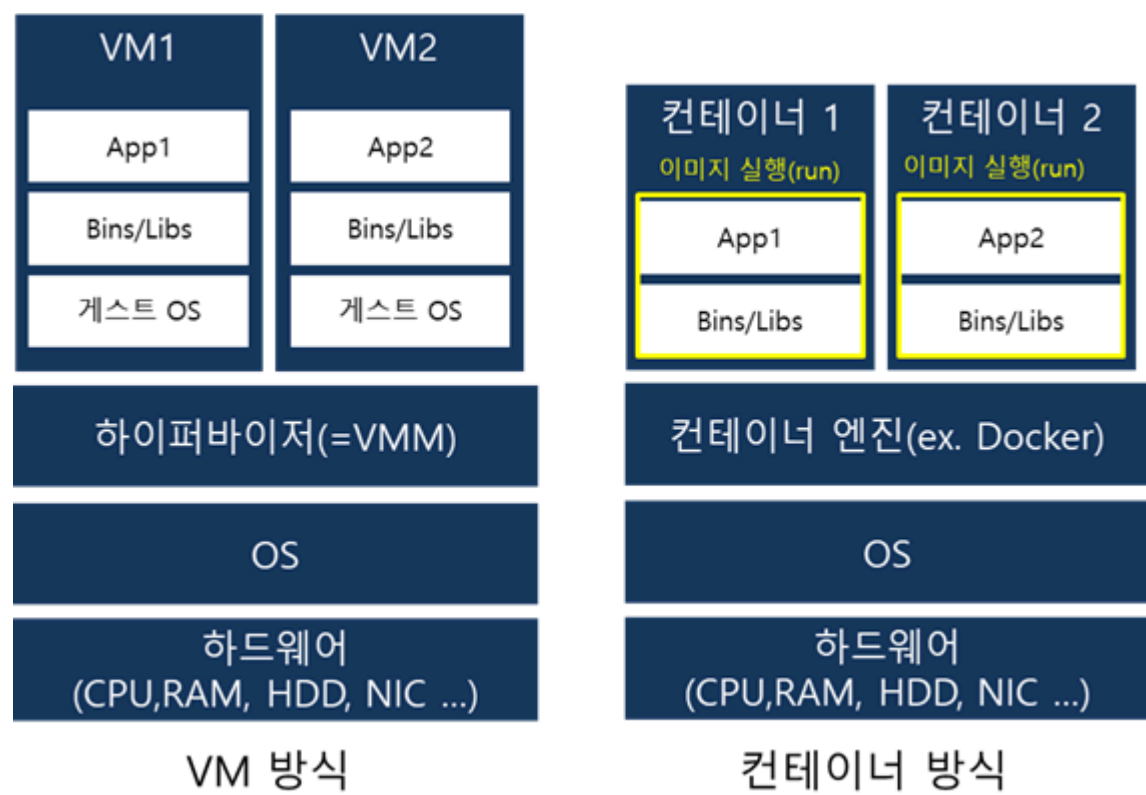
## 컨테이너 환경을 제공하는 기술

- chroot
- Docker
- LCX
- Solaris Container(Zones)
- FreeBSD Jail
- WPARS(AIX)
- rkt

# 왜 (도커) 컨테이너를 써야하나요?

## 컨테이너 vs 가상머신

가상머신은 애플리케이션을 동작시키기 위해서 애플리케이션이 사용하는 리소스만 사용하는 것뿐만 아니라 운영체제가 동작하기 위한 리소스가 추가로 필요합니다. 그러나 컨테이너는 애플리케이션이 동작하기 위한 리소스만 사용하기 때문에 훨씬 더 빠르고 가볍게 동작시킬 수 있습니다.



서버 운영 측면에서 배포와 운영에 도커를 꼭 써야만 하는 건 아닙니다.  
하지만 수평적 확장의 자유로움, 서버의 견고함을 보장하면서 동적으로 바꿀 수 있는 유연함, 인수인계 시간 절약 등의 편리한 해결 방법을 제공하는 도커를 쓰지 않을 이유는 없습니다.