

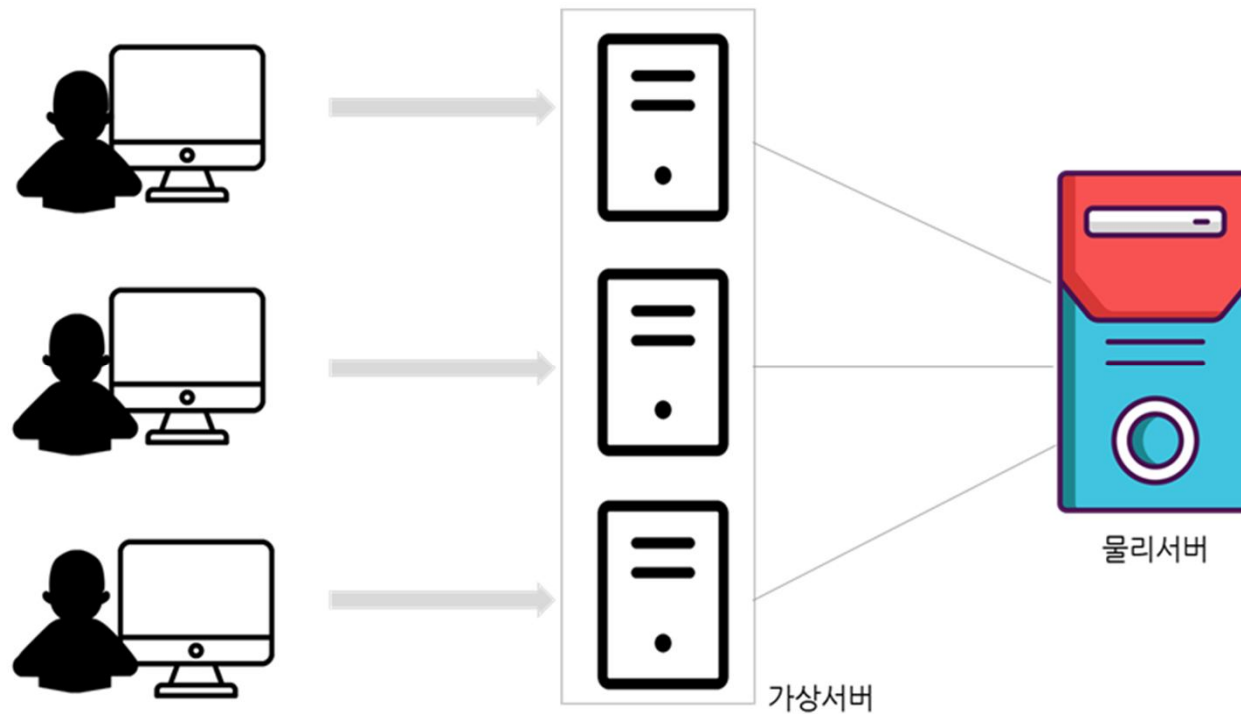
클라우드를 실현하는 기술들

IaaS를 구성하는 주요기술	서버 가상화 기술	하이퍼바이저형, 컨테이너 형 등
	네트워크 가상화 기술	VLAN, VPN, NFV, SDN 등
	스토리지 기술	블록 스토리지, 파일 스토리지 등
	가상화 기반 기술	OpenStack, Vmware,, Vsphere 등
PaaS를 구성하는 주요기술	데이터베이스 기술	RDB, NoSQL 등
	분산처리 기술	Apache Hadoop, Apache Spark 등

1.1 가상화(virtualization) 개요

- 실체가 없는 것을 마치 존재하는 것처럼 보이는 기술
- 시스템이나 네트워크를 구성하는 각종 물리 자원을 논리적으로 마치 다른 자원인 것처럼 인식 시키는 기술
- 대표적인 가상화 기술
 - 자원분할 : 단일 물리 자원을 논리적으로 분할하여 별도의 물리자원으로 인식, 동작시키는 기술
 - 자원통합 : 복수의 물리자원을 논리적으로 통합하여 단일한 물리 자원처럼 동작시키는 기술
- 특징
 - 가상화의 유연성
 - 가용성 향상

(예) 물리서버 1대에 여러 대의 서버가 있는 것처럼 구축, 물리서버를 늘리지 않고 필요에 따라 서버를 늘리거나 줄일 수 있음(자원 분할 기술)

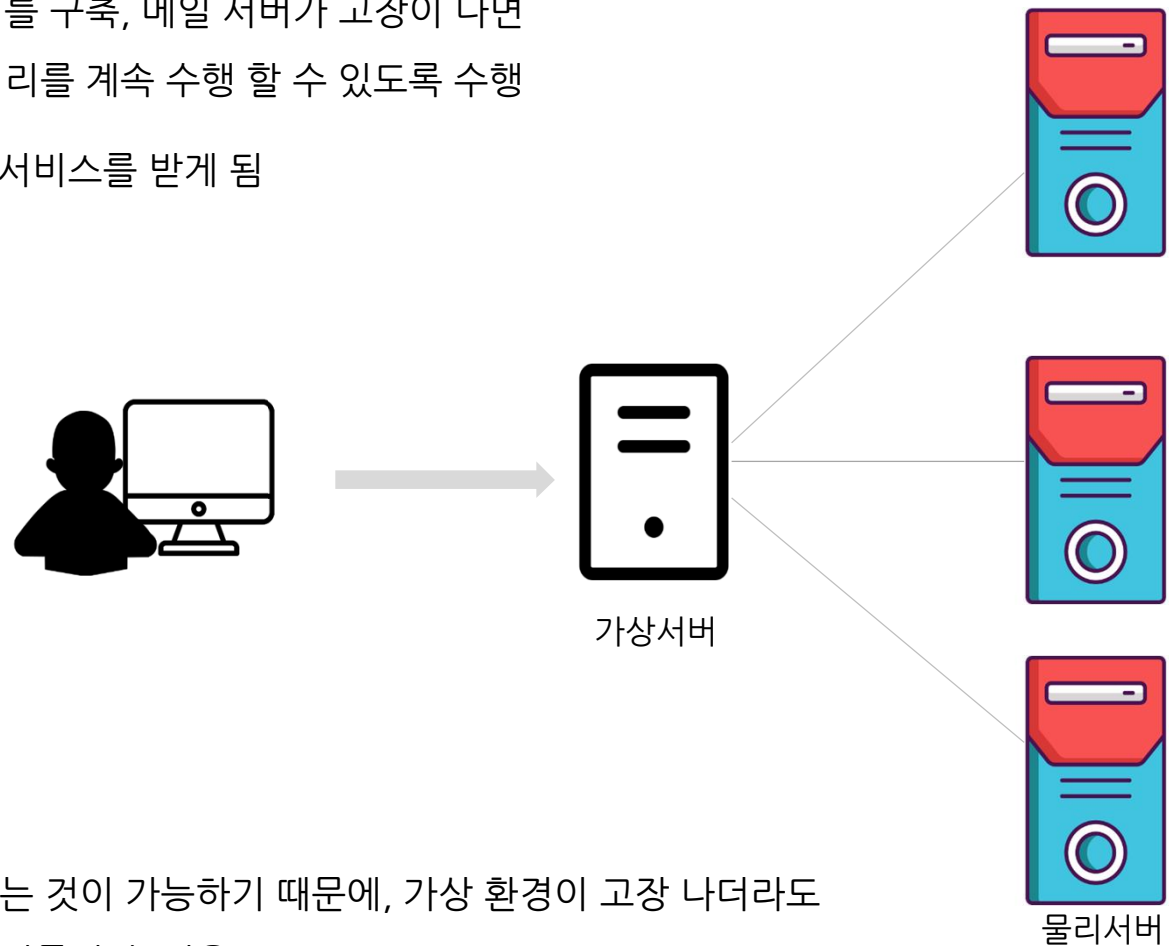


· 가상화의 유연성

- 가상화를 통해 물리적 환경의 제약에 얽매이지 않고 유연하게 환경을 구축 가능

(예) 물리서버가 여러 대 구축되어 있지만 사용자에게 한 대의 서버로 보이게 하는 기술(자원통합기술)

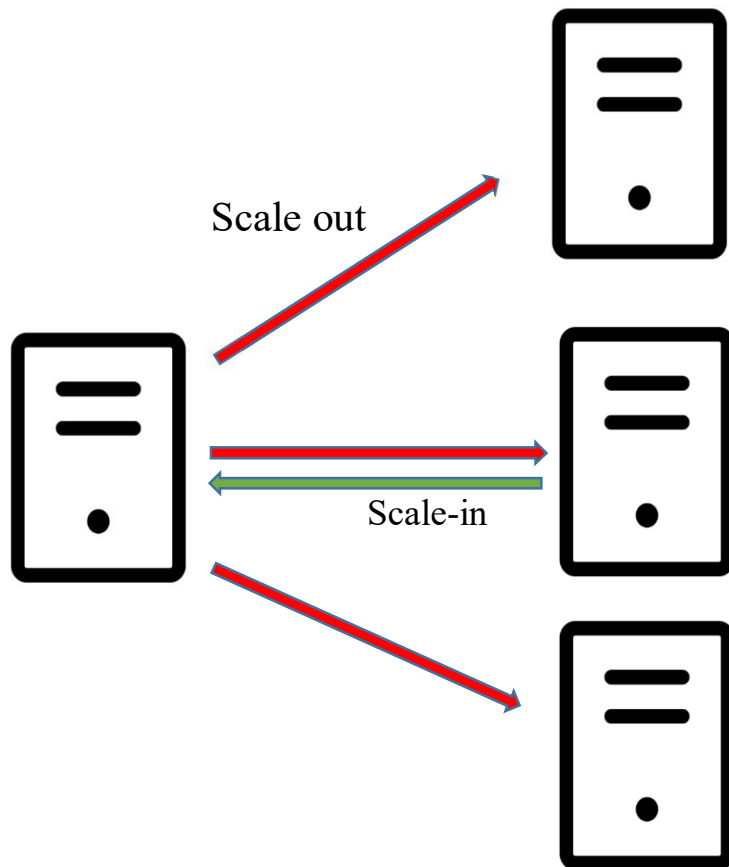
- Gmail Server : 구글을 여러 대의 메일 서버를 구축, 메일 서버가 고장이 나면
다른 메일 서버에서 같은 처리를 계속 수행 할 수 있도록 수행
- 사용자는 서버가 고장 난 것을 인지 못하고 서비스를 받게 됨



· 가용성 향상

- 생성한 가상 환경을 각각 환경별로 백업해 두는 것이 가능하기 때문에, 가상 환경이 고장 나더라도 백업을 바탕으로 복원해 곧바로 이전 상태로 되돌릴 수 있음

[참고] Scale-out/Scale-in



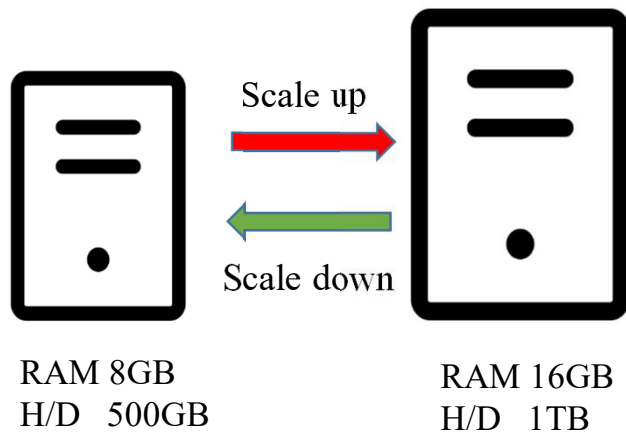
Scale Out

- 장비를 추가해서 확장하는 방식
- 기존 서버만으로 용량이나 성능의 한계에 도달했을 때, 비슷한 사양의 서버를 추가로 연결
- 데이터 용량이 증가할 뿐만 아니라 기존 서버의 부하를 분담해 성능 향상의 효과를 기대

(예) Scale out : 서버 수 증가 (ex. 1대 → 3대)

Scale in : 서버 수 감소 (ex. 3대 → 1대)

[참고] Scale-up/Scale-down



Scale up

- 기존의 서버를 보다 높은 사양으로 업그레이드하는 것
- 서버 자체의 스펙을 증강하는 것
- 성능이나 용량 증강을 목적으로 하나의 서버에 디스크를 추가하거나 CPU나 메모리를 업그레이드시키는 것

(예) Scale up : RAM 용량 증가 (ex. 8GB → 16GB)

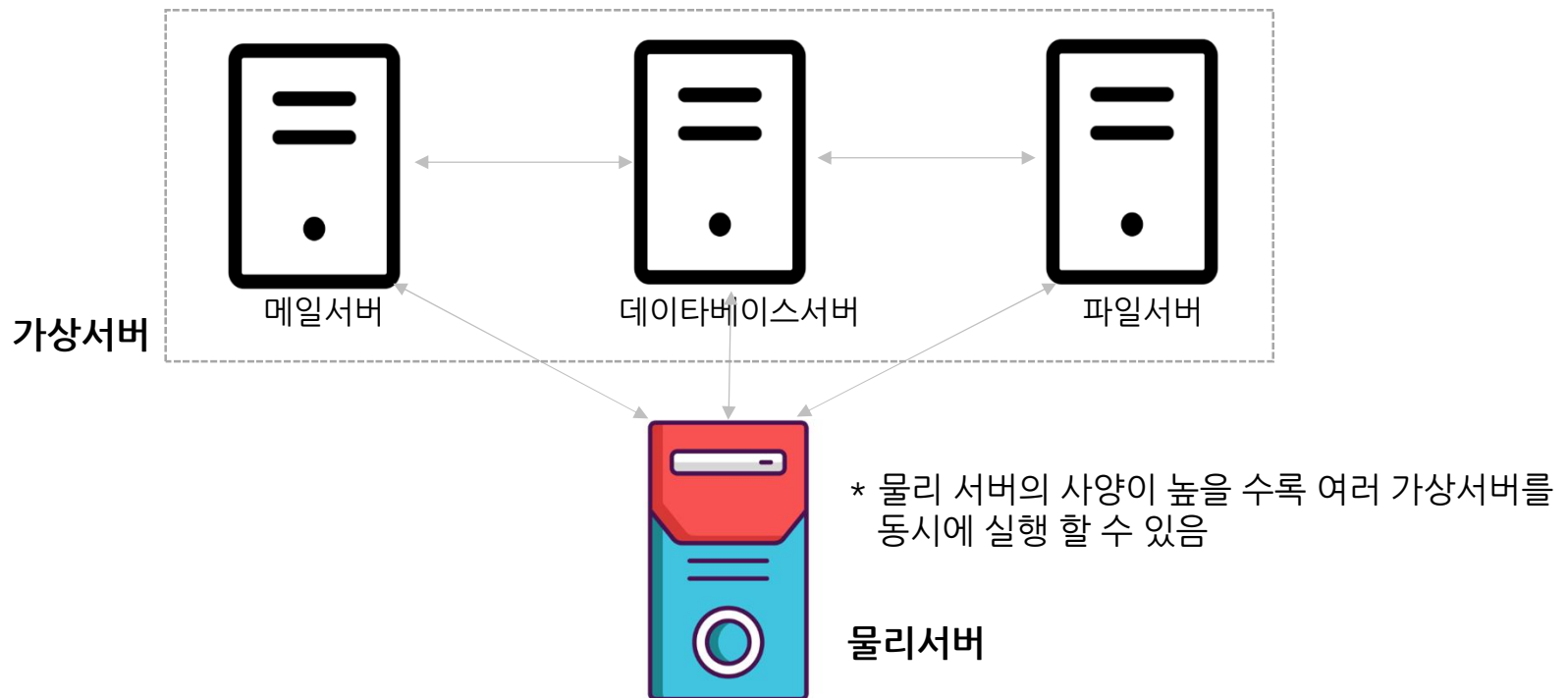
Scale in : RAM 용량 감소 (ex. 16GB → 8GB)

1.2 가상화 종류

- 서버 가상화
- 네트워크 가상화
- 스토리지 가상화
- 데스크탑 가상화

1) 서버 가상화

- 물리서버를 가상화 하는 것, 물리 서버 하나에 가상 서버를 여러 대 서버 환경을 구축하는 기술
- 가상 서버에는 CPU, 메모리, 스토리지, 네트워크 등이 에뮬레이트 되므로 물리서버 처럼 사용
- 각각의 가상 서버는 독립된 상태이므로, 같은 물리 서버에 있는 가상 서버 중 하나가 바이러스와 같은 위협에 노출되더라도 다른 가상 서버에 영향을 미치지 않음

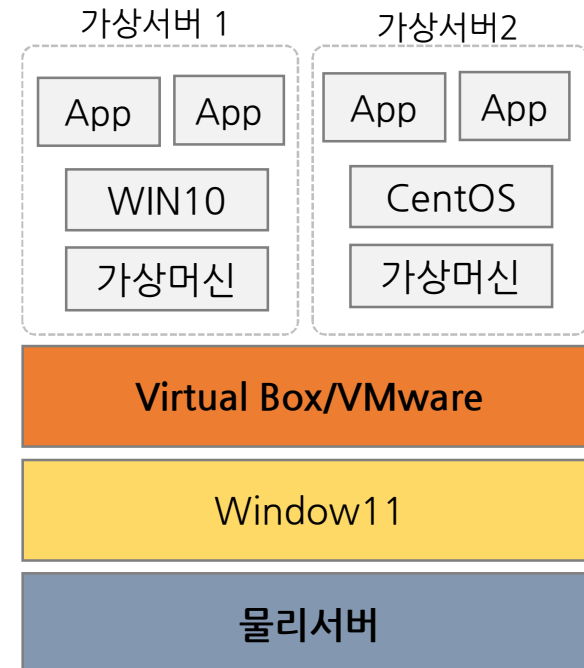
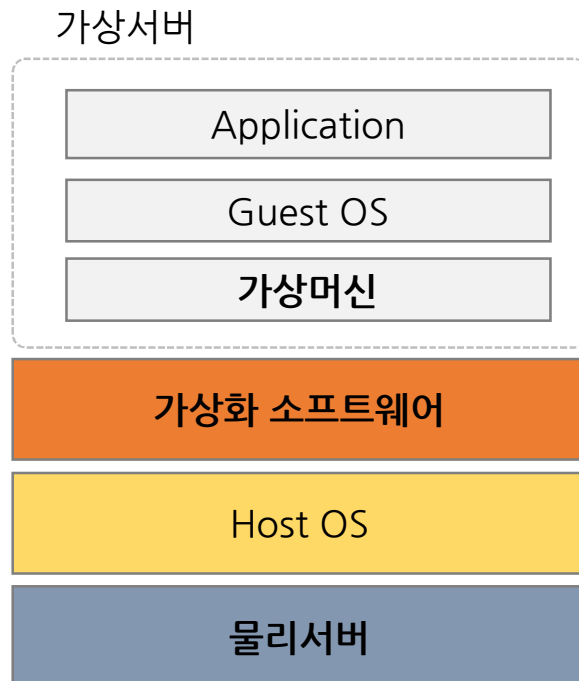


2) 서버 가상화 구축 유형

- 호스트 OS형 가상화
- 하이퍼바이저형 가상화
- 컨테이너형 가상화

① 호스트 OS 형 가상화

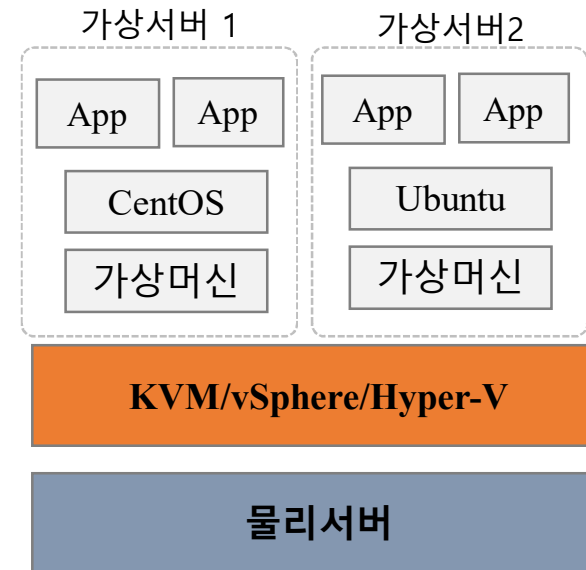
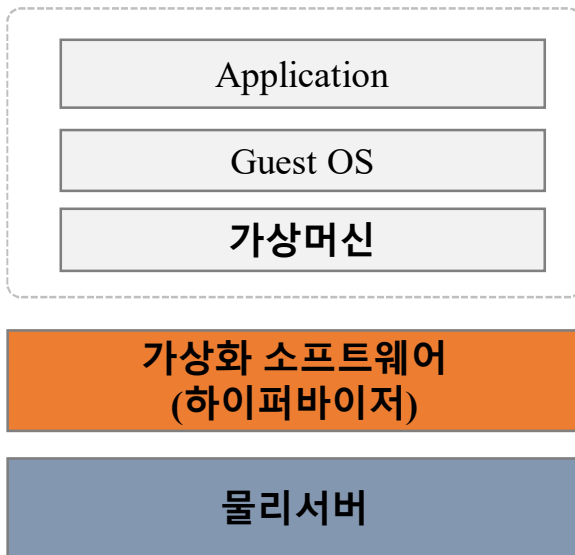
- 물리서버에 OS가 설치되어 있으며, 호스트 OS 위에 가상 서버를 실행하는 기술
- 가상화 소프트웨어 : VMWare Workstation, Virtual Box, VMware Fusion 등



② 하이퍼바이저 가상화

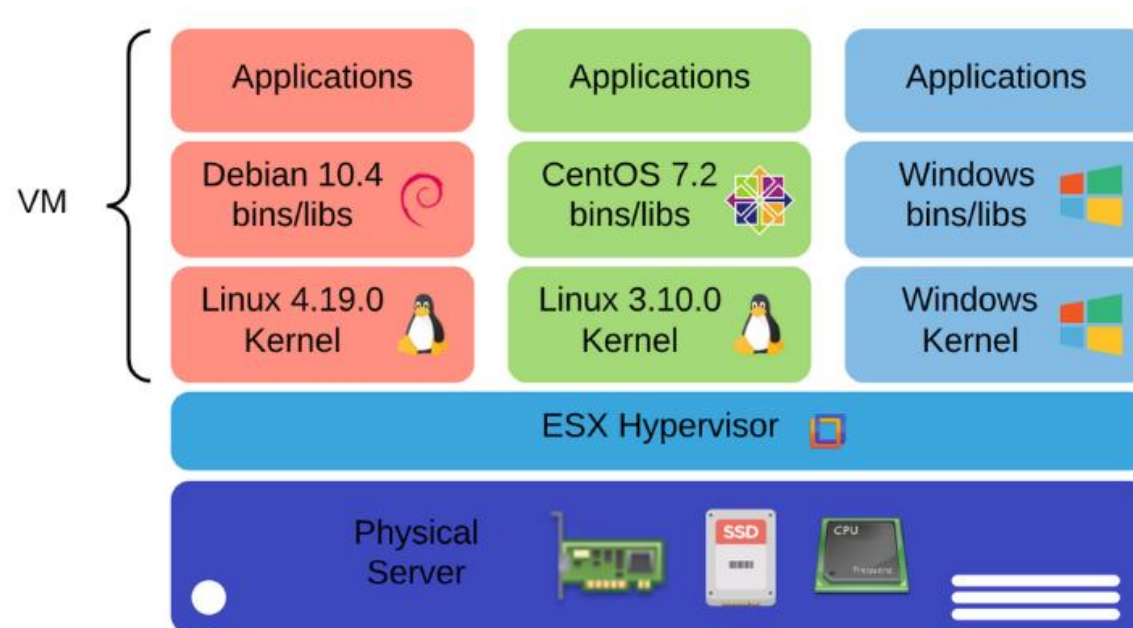
- 하이퍼바이저라는 가상 환경을 관리하는 소프트웨어를 물리서버에 설치
- 물리서버에 OS를 설치할 필요가 없음
- 하이퍼바이저 가상화 소프트웨어 : Hyper-V, VMWare vSphere, Xen, KVM

가상서버



② 하이퍼바이저 가상화

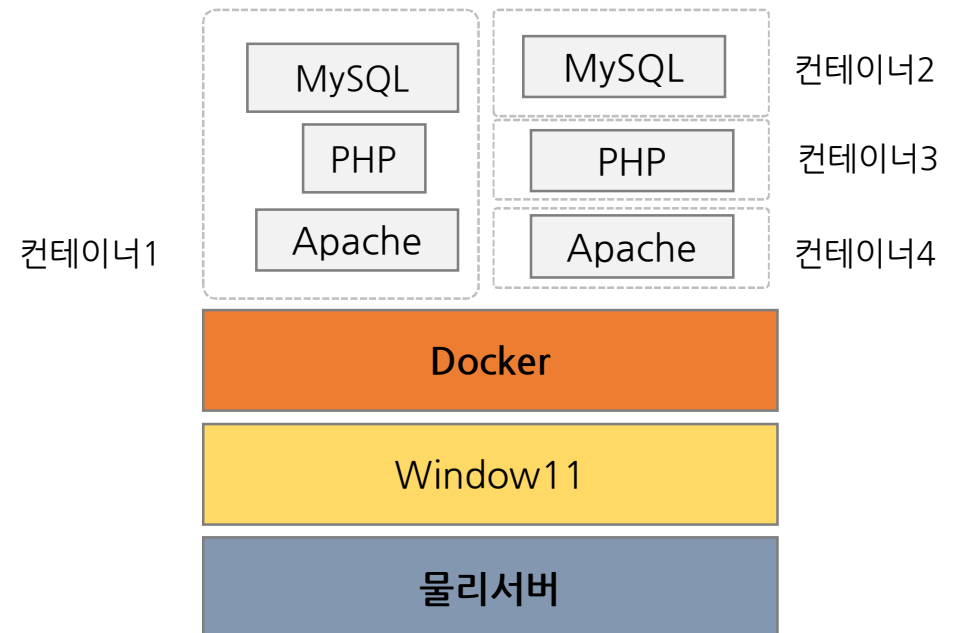
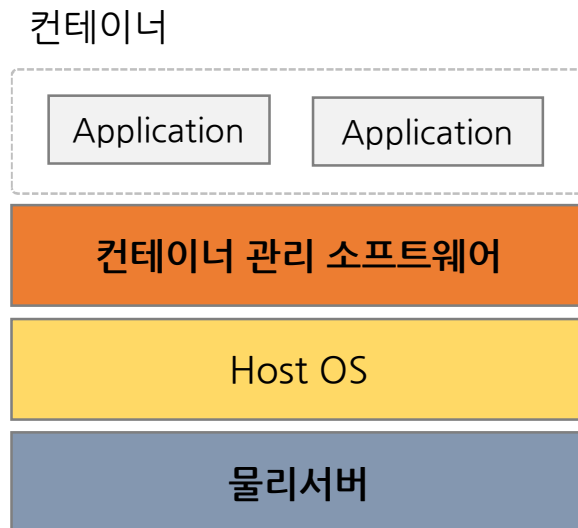
- 비교적 규모가 큰 가상 환경을 구축하는데 사용
- 호스트 OS가 없고 게스트OS에서 직접 제어하기 때문에 컴퓨터의 리소스를 활용하기 쉬움
- 처음부터 가상 환경을 구축해야 한다면 하이퍼바이저형 가상화를 선택하는 경우가 많음
 - 이미 서비스 중인 물리서버가 있는 경우에는 호스트 OS 형 가상화를 많이 사용



Bare metal hypervisor (가상화 타입 1)

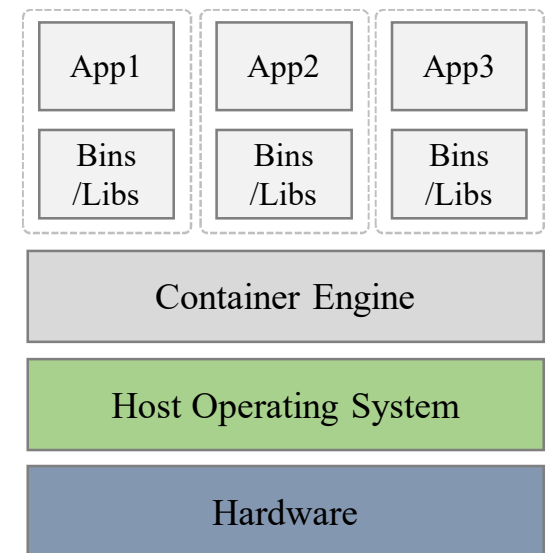
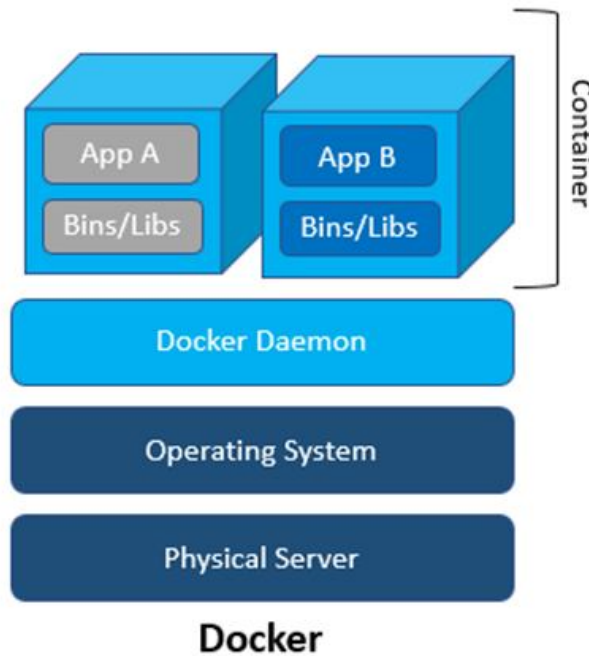
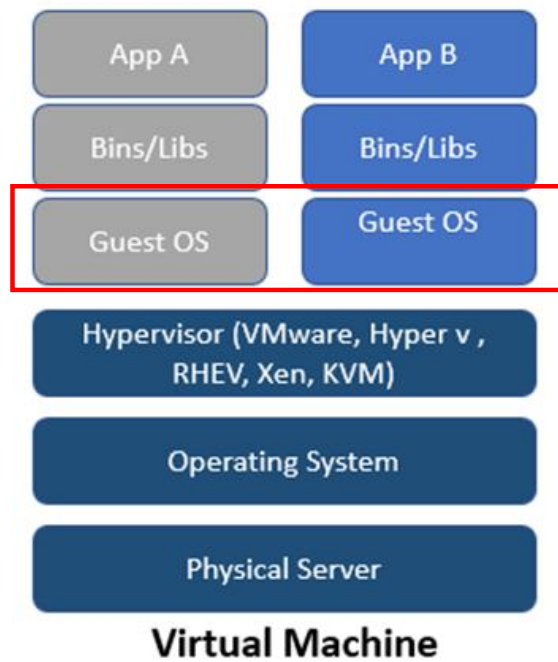
③ 컨테이너 가상화

- 컨테이너라는 애플리케이션과 실행환경을 분리하는 방식으로 OS 단위가 아닌 애플리케이션 단위로 가상화하는 기술
- 여러 컨테이너를 조합해 환경을 구축 할 수 있으므로 유연하게 환경을 구축할 수 있음
- 설치 같은 번거로움을 줄일 뿐만 아니라 가상 하드웨어나 게스트OS를 실행할 필요가 없어 성능도가 좋아짐
- 컨테이너형 가상화 소프트웨어 : Docker



컨테이너(container)

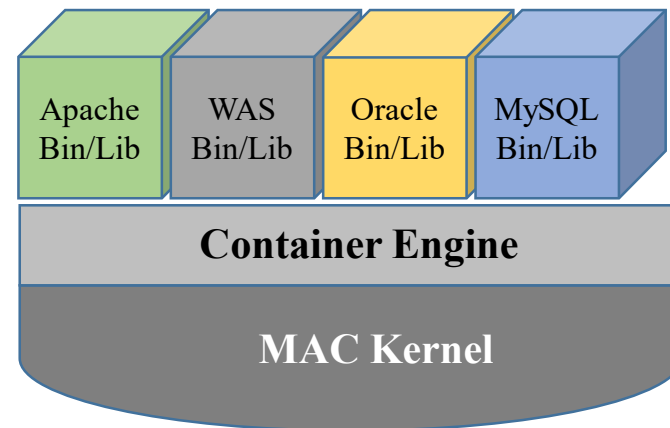
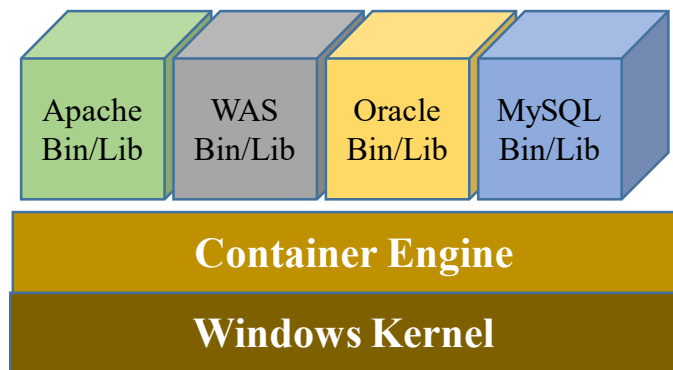
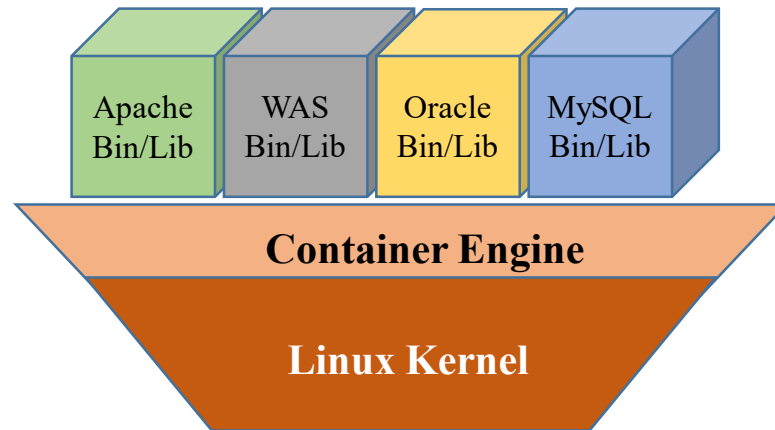
- Host OS로부터 격리된 환경에서 실행되는 프로세스 그룹
- 호스트 OS상에서 논리적으로 구역(컨테이너)을 나눔

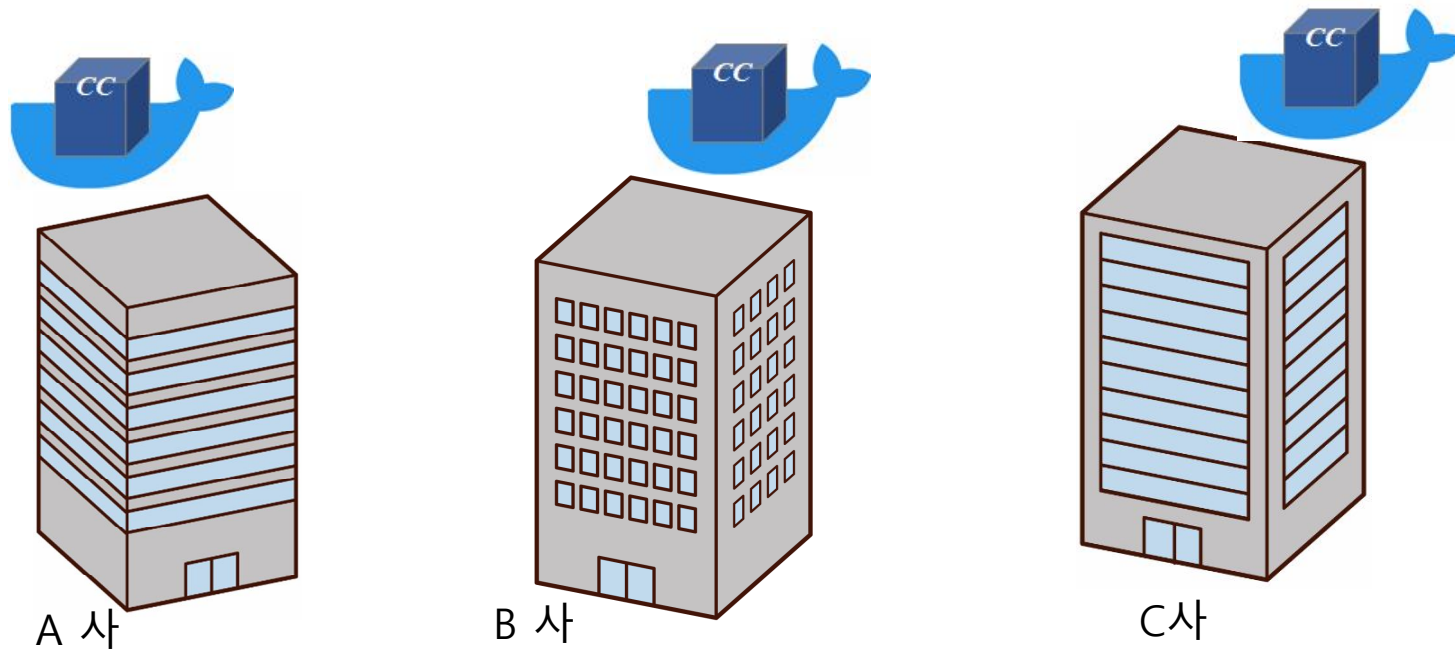
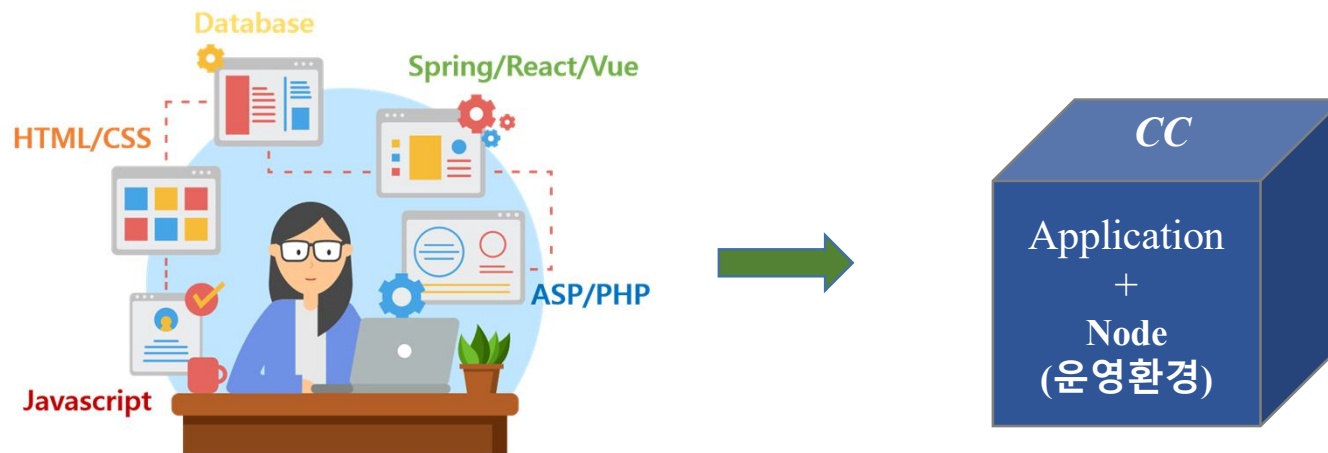


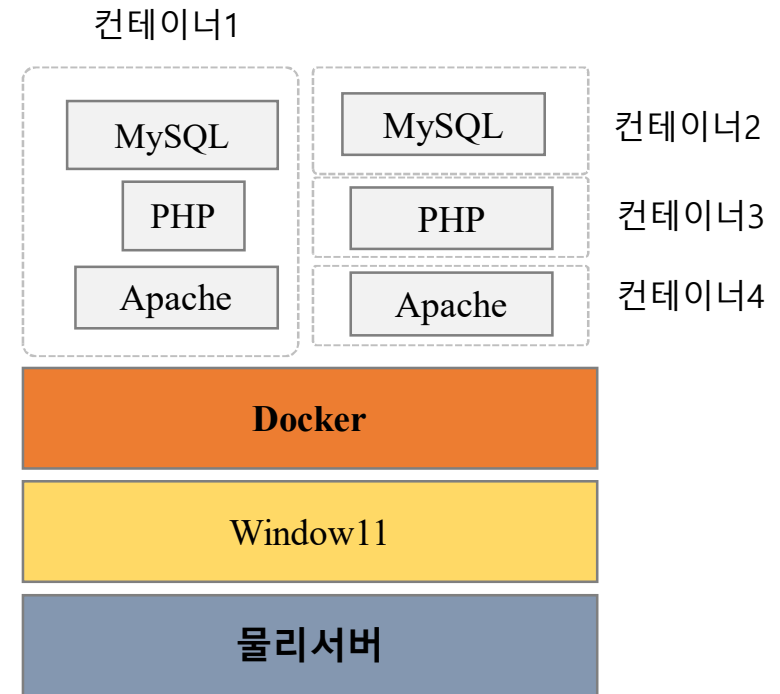
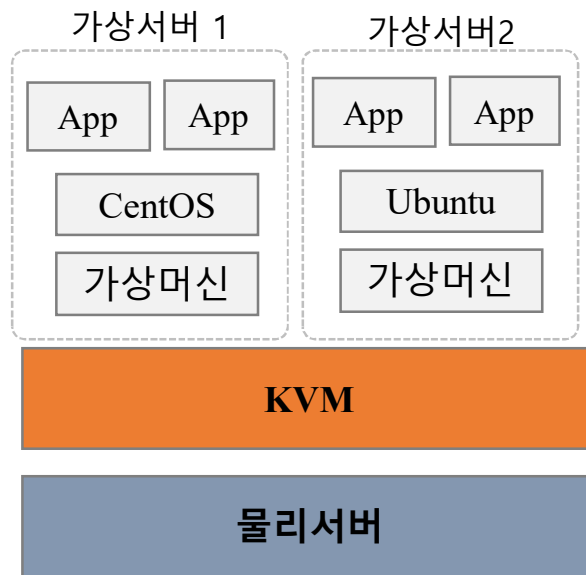
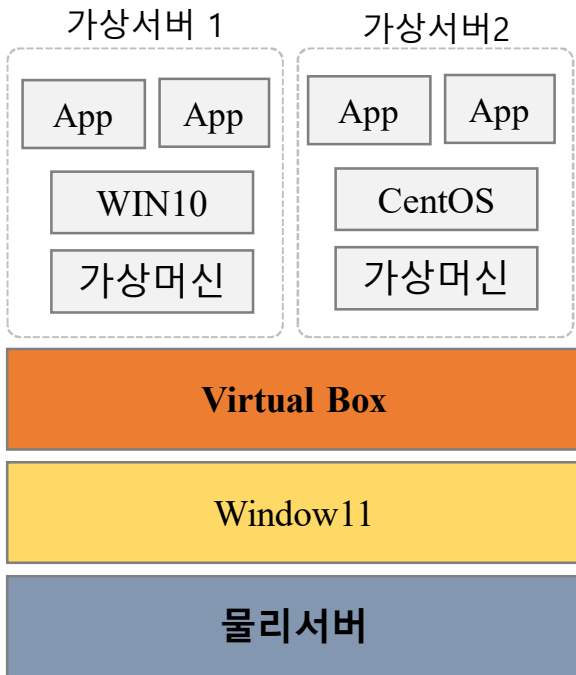
컨테이너 가상화 소프트웨어 (컨테이너 엔진)

	LXD	Docker	LXC	OpenVZ	Virtuozzo
Release	2015	2013	2008	2005	2001
개발사	Canonical	Docker Inc.	IBM, Parallels, Canonical	OpenVZ Community	Parallels
지원OS	Linux	Linux, Windows, MacOS	Linux	Linux	Linux, Windows
가격	무상	무상	무상	무상	유상
형태	오픈소스	오픈소스	오픈소스	오픈소스	독점









3) 가상화 목적과 장점

① 자원 사용의 효율성을 높이기 위해

- 여러 서버를 모아서 자원의 사용률을 개선
- 풀(pool) 개념을 사용 함으로서 자원에 대한 빠른 전개와 회수가 가능
- 애플리케이션들을 완전히 격리함으로써 멀티테넌시 환경을 구성

*** 멀티테넌시 (multitenancy) : 서로 다른 고객이 서버 리소스를 나누어 사용하는 공유 호스팅**

② 가상화의 유연성

- 물리적 환경의 제약에 얽매이지 않고 유연하게 환경을 구축 할 수 있음

③ 가용성 향상

- 가용성은 시스템을 장애 없이 계속 사용할 수 있는지 여부를 나타내는 지표
- 가상서버는 백업이 가능하므로 장애가 발생하기 전 상태로 되돌릴 수 있음
- 이로 인해 시스템의 가용성이 향상됨
- 가상서버는 물리 서버의 전원이 다운(down)되면 사용 할 수 없음

4) 가상화 단점

- 가상 서버 단위의 가동률뿐만 아니라 물리 서버의 가동률도 고려해야 함
 - 물리서버가 움직이지 않으면 당연히 가상서버도 움직이지 않기 때문
 - 물리서버에 장애가 발생하면 해당 서버에서 실행되는 모든 가상서버에 영향을 줄 수 있음
- 가상화를 도입하거나 유지하는데 많은 비용이 듦
 - 소규모 서비스나 장애가 발생했을 때 업무 영향이 작은 시스템 무리하게 가상화를 사용할 필요가 없음
 - 가상화 구축에 필요한 비용이 제공 서비스에 들어가는 비용과 비교해 효율적인지 아닌지 가상화를 도입하기 전에 충분히 검토해야 함