

지상편



일프로의 한방정리

쿠버네티스 어나더 클래스

가상화 한방정리

Sprint 3



가상화 (Virtualization)

어플리케이션 (Application)

데이터 (Data)

런타임 (Runtime)

미들웨어 (Middleware)

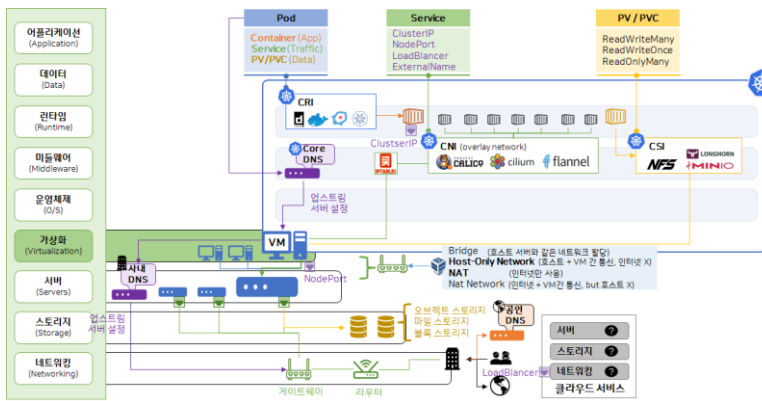
운영체제 (O/S)

서버 (Servers)

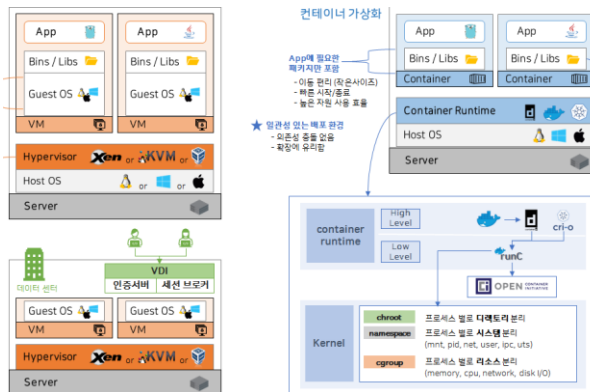
스토리지 (Storage)

네트워킹 (Networking)

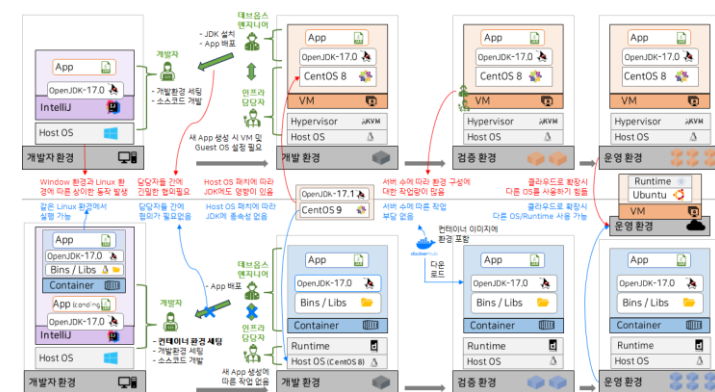
쿠버네티스에서 가장 어려운 리소스



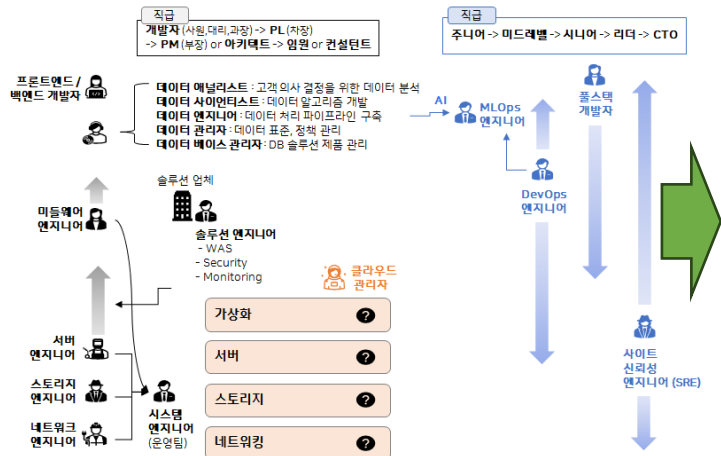
누구나 알고 있는 가상화 기술 비교



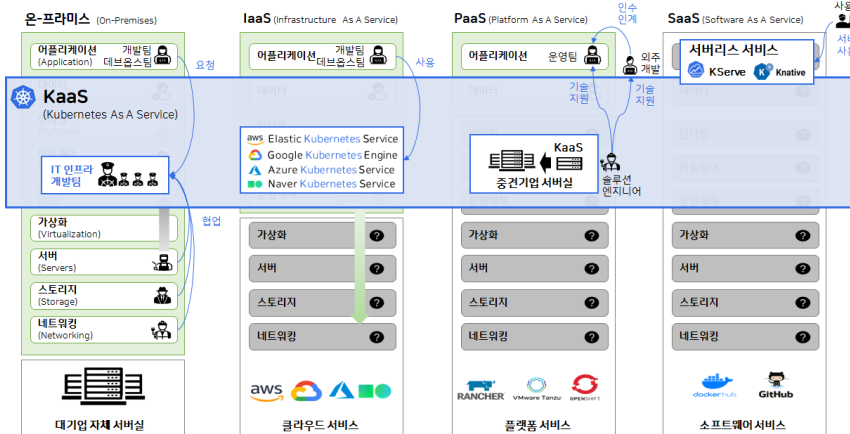
아무나 모르는 컨테이너 기술의 장점



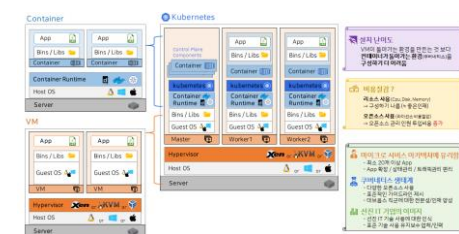
IT에는 어떤 직군들이 있을까?

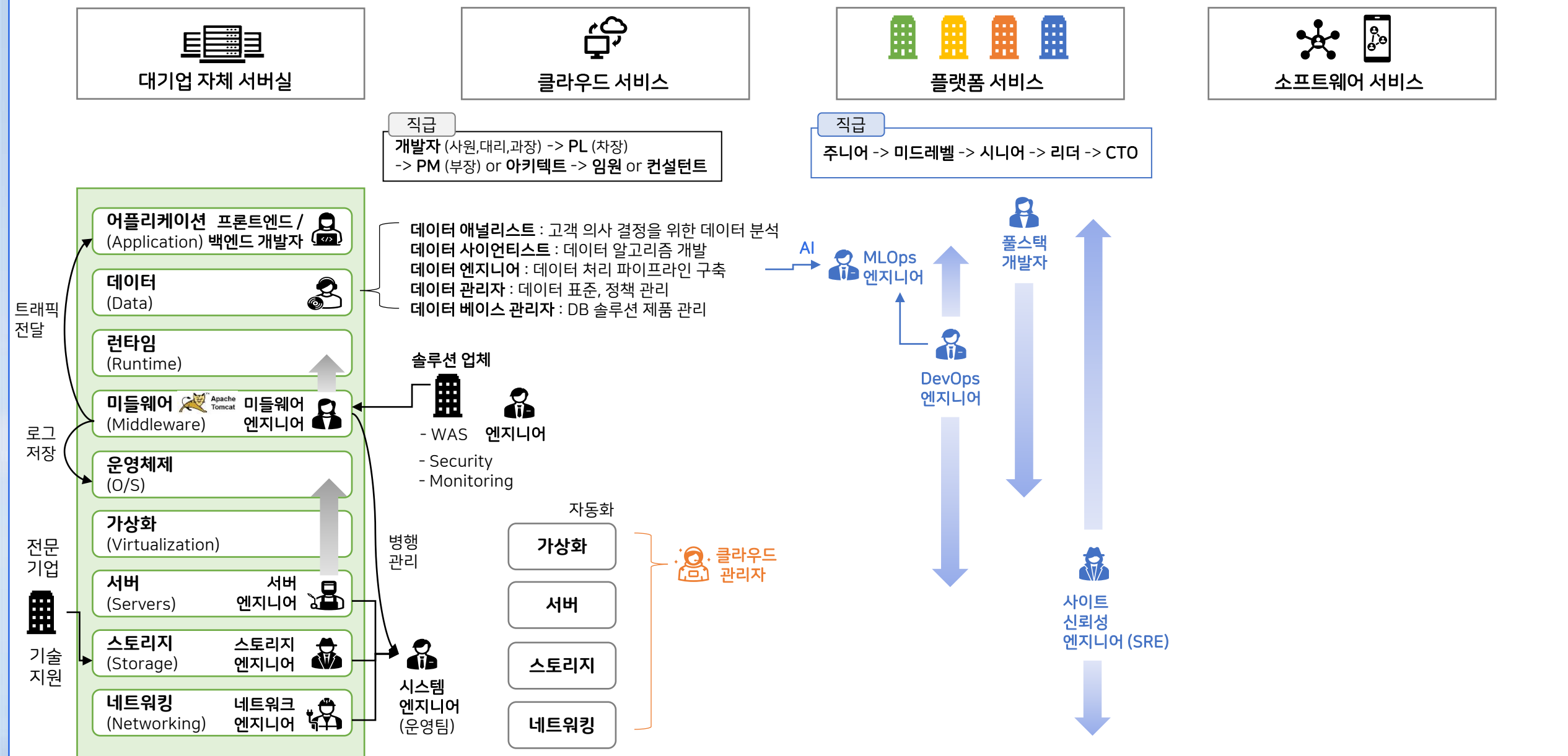


한번쯤 알고 넘어가면 좋을 서비스 모델



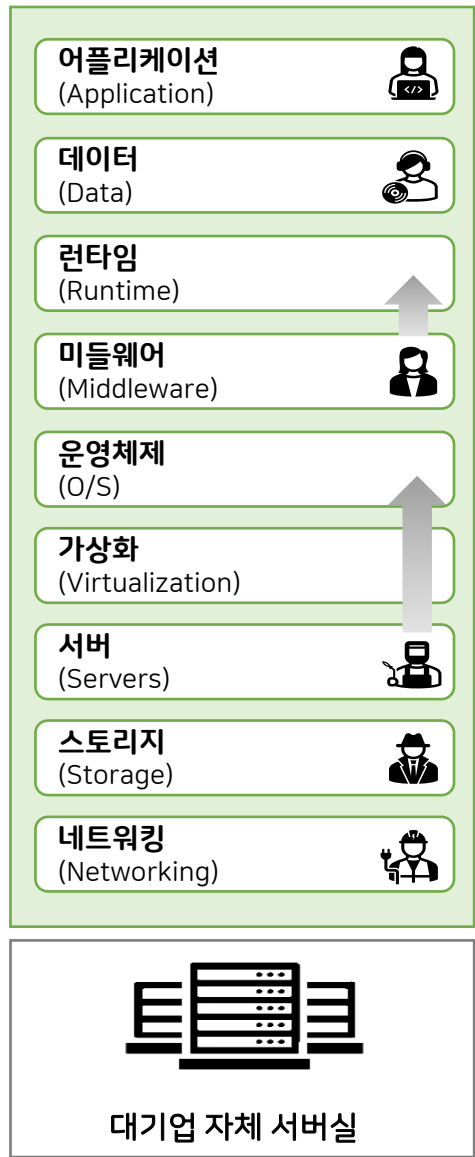
쿠버네티스의 단점







온-프라미스 (On-Premises)



IaaS (Infrastructure As A Service)

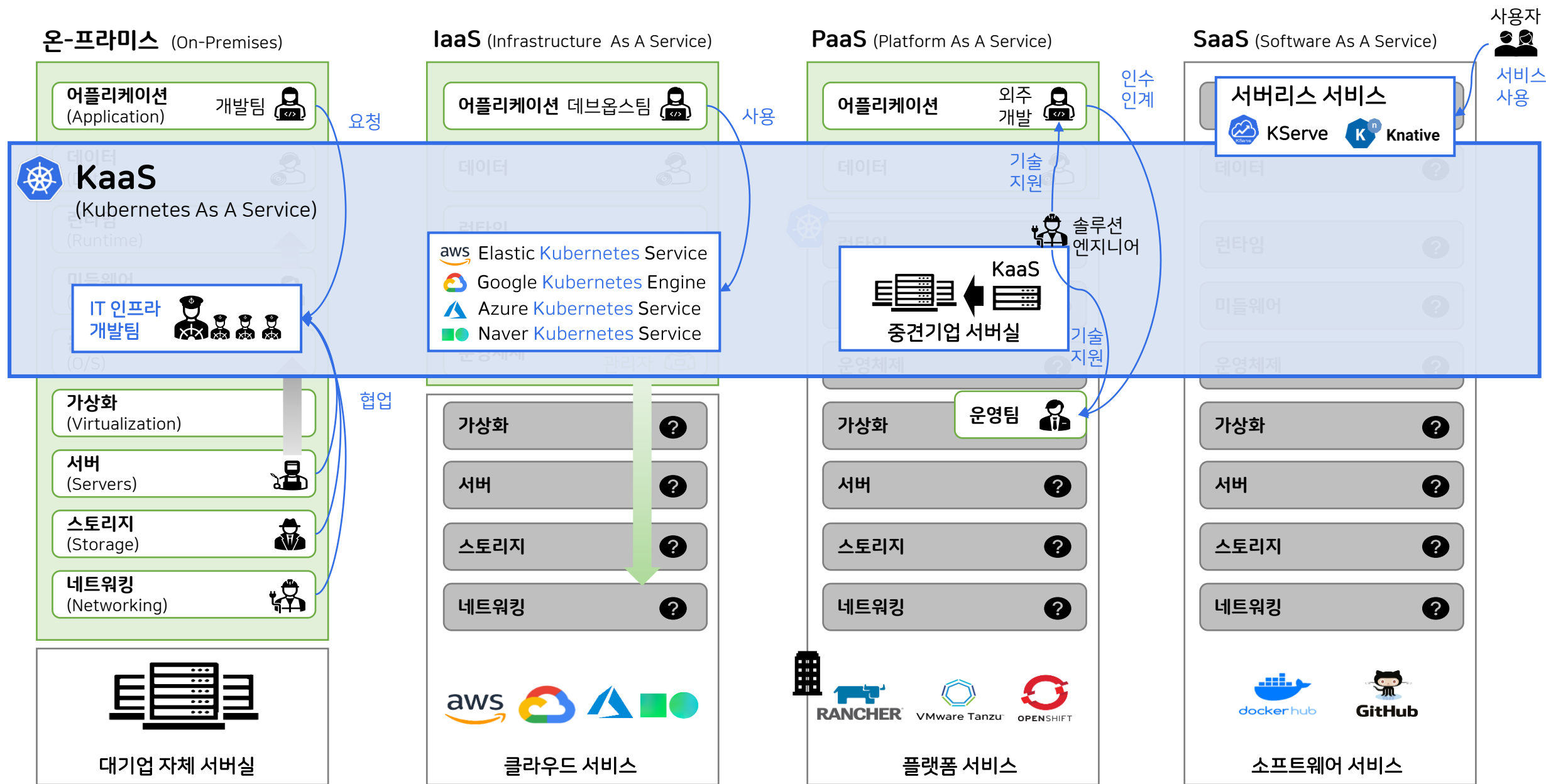


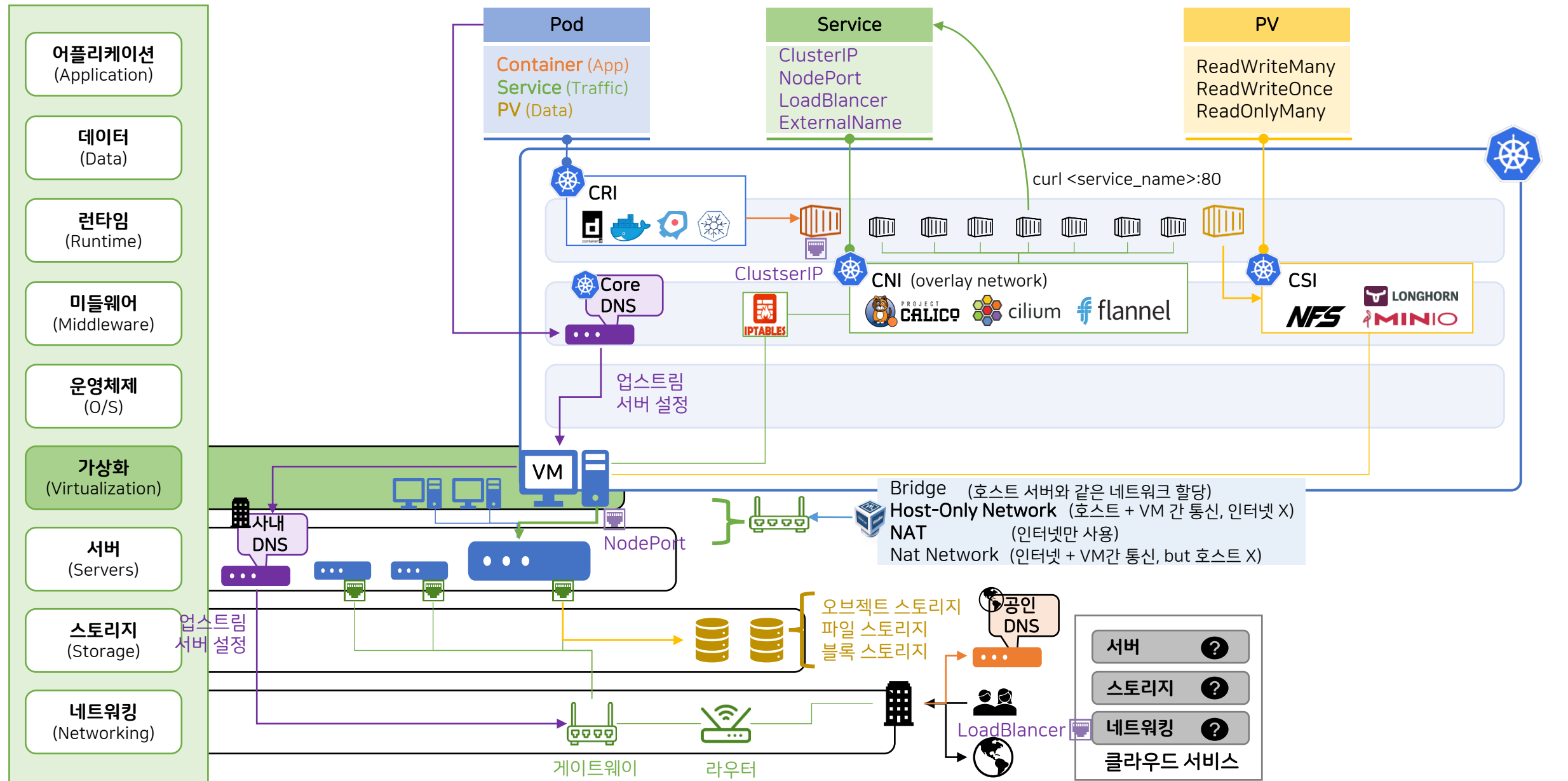
PaaS (Platform As A Service)



(test) Environment As A Service
DB As A Service
SaaS (Software As A Service)





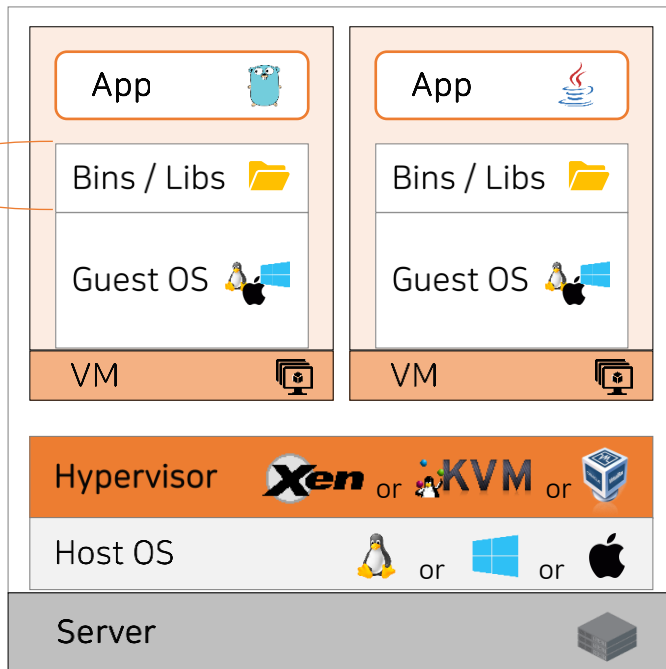




호스트 가상화

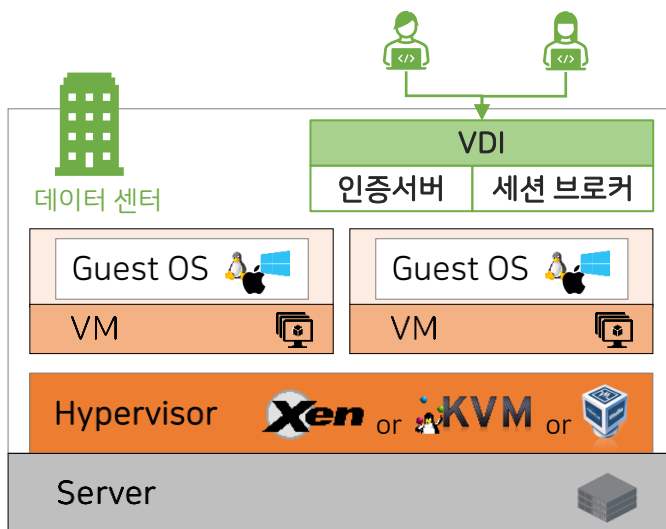
App + OS 관련 패키지

- 큰 사이즈
- GuestOS를 위한 자원 사용량 부담



하이퍼바이저 가상화

- 데이터 센터에서 주로 사용하는 가상화 환경
- 보안을 위한 망 분리 목적



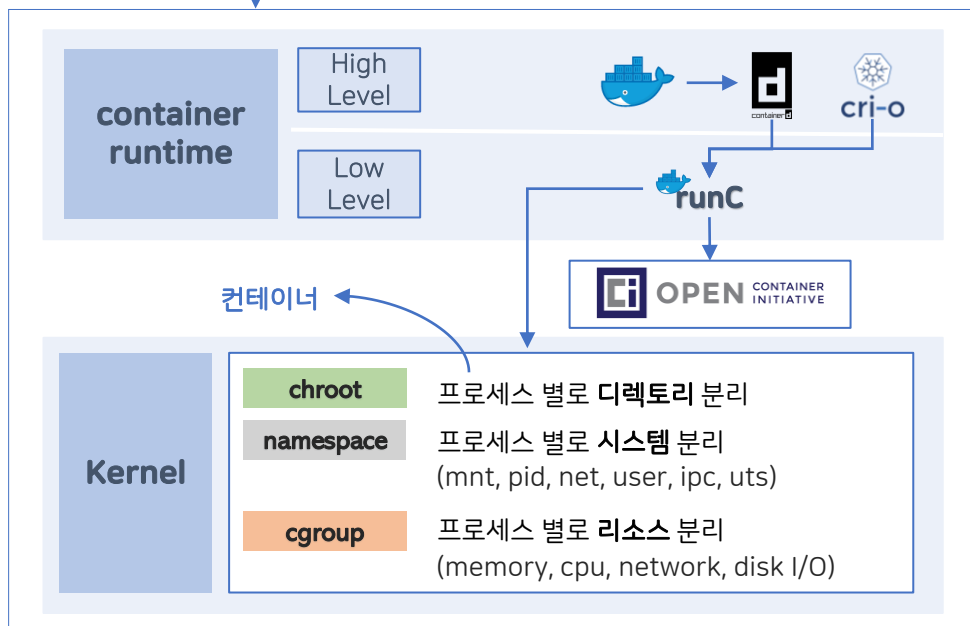
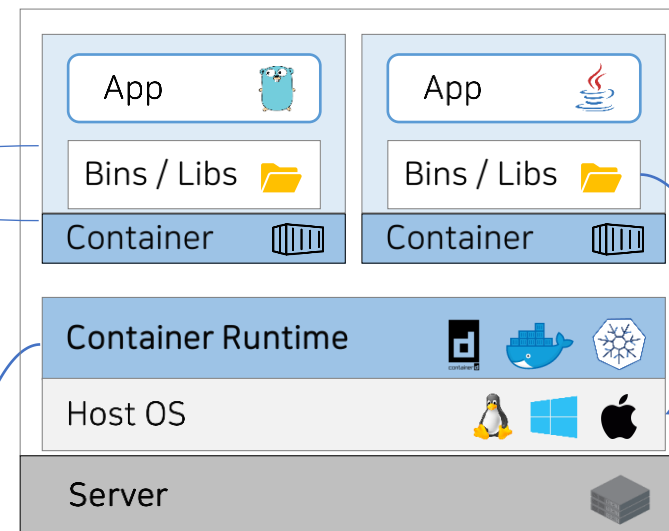
컨테이너 가상화

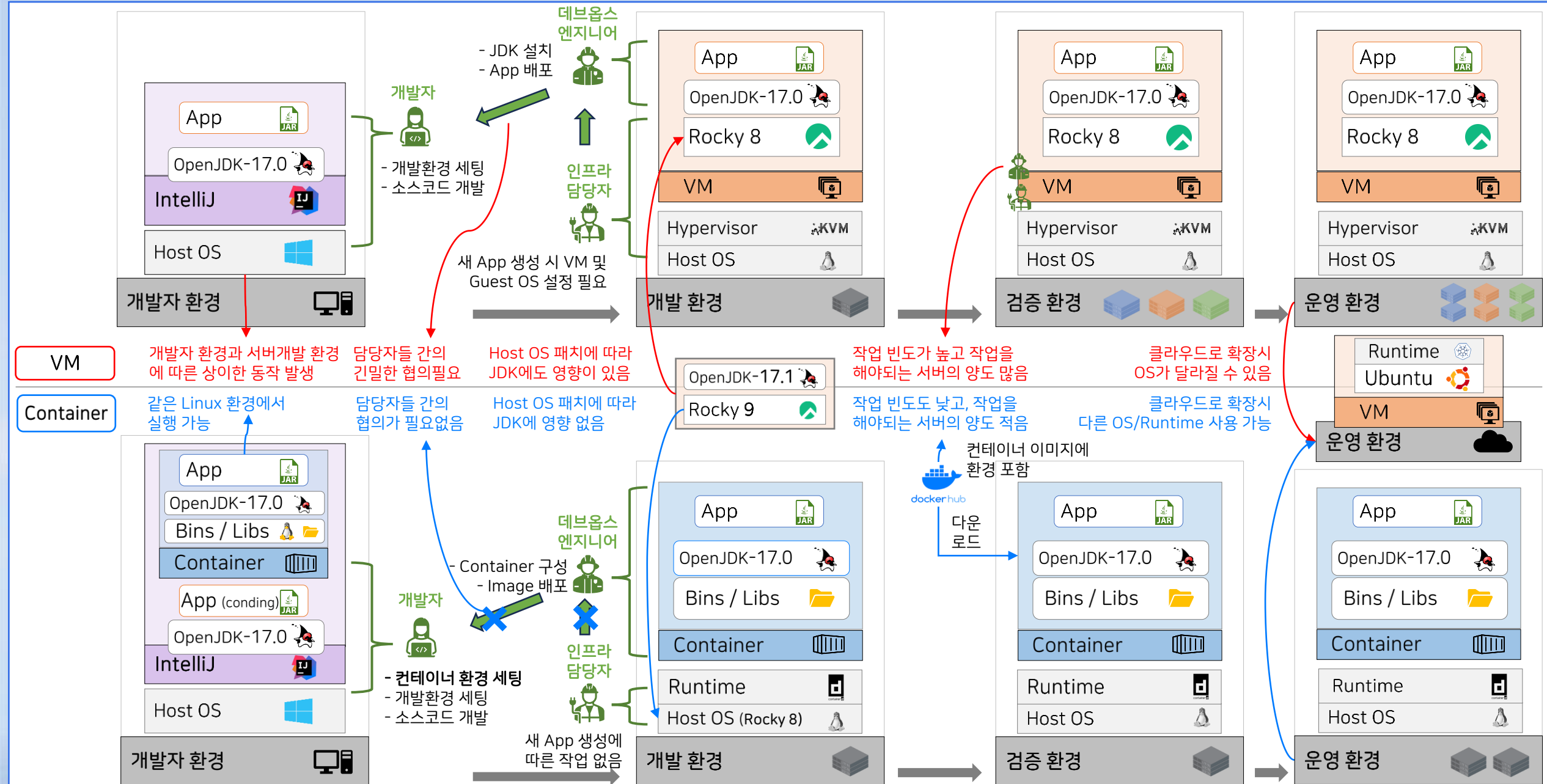
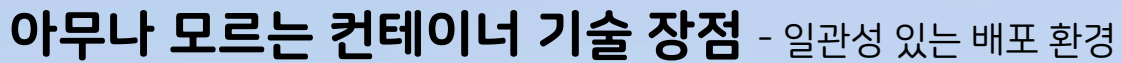
App에 필요한 패키지만 포함

- 이동 편리 (작은 사이즈)
- 빠른 시작/종료
- 높은 자원 사용 효율

★ 일관성 있는 배포 환경

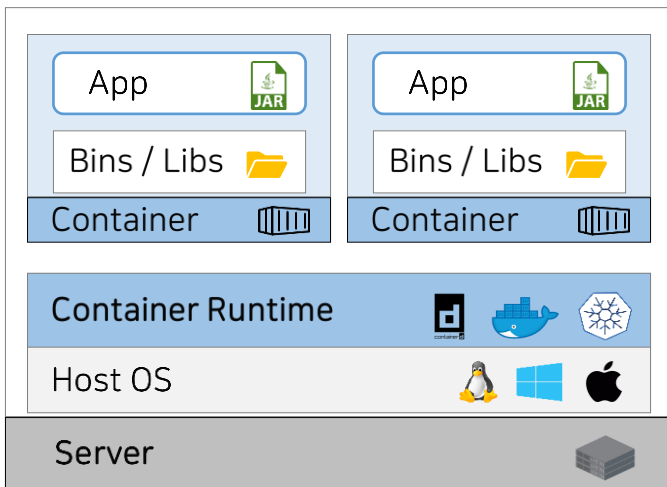
- 의존성 충돌 없음
- 확장에 유리함



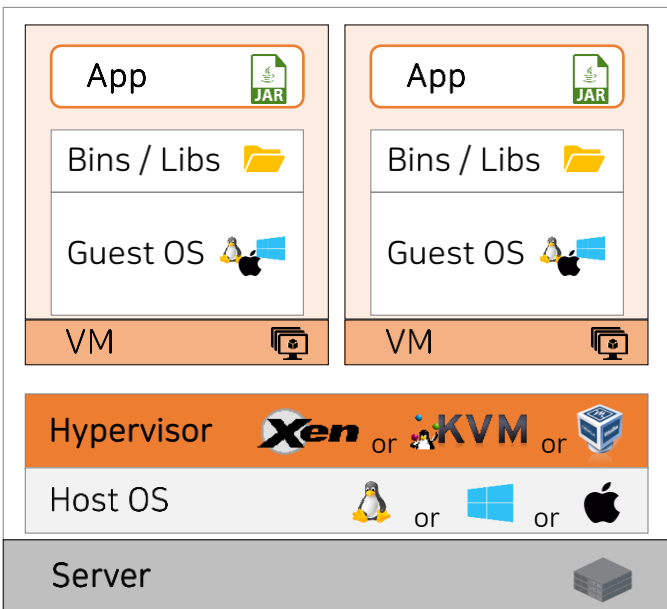




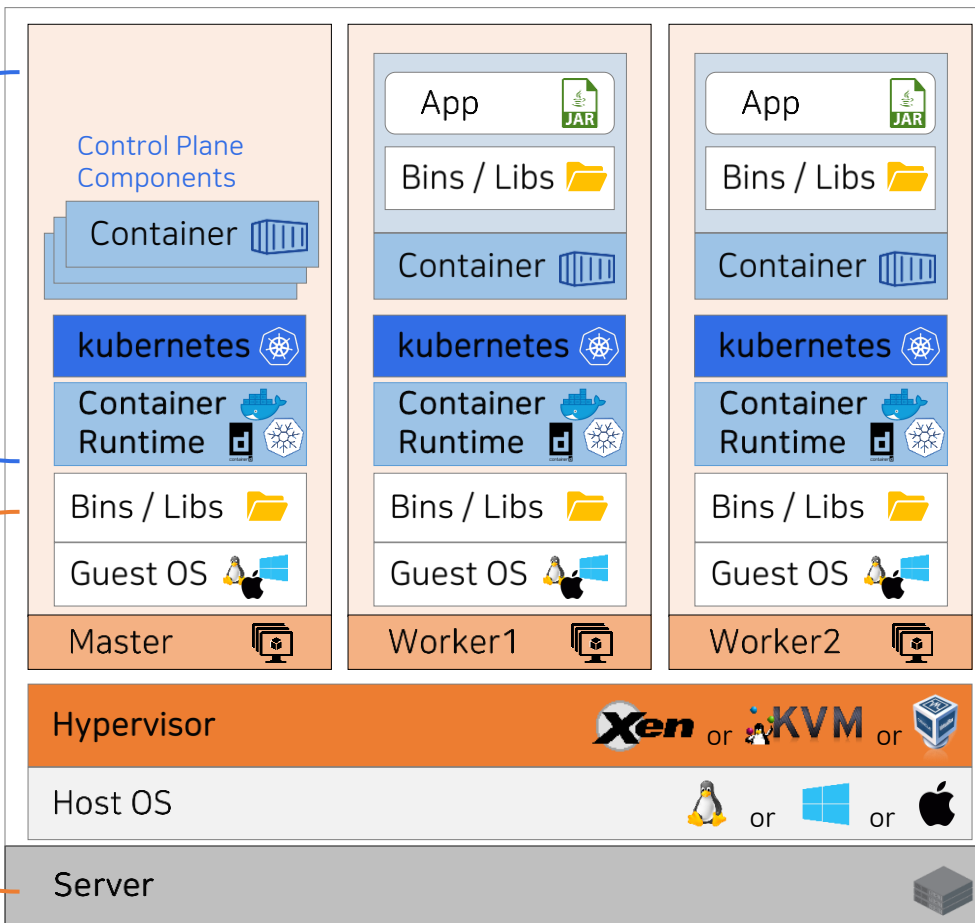
Container



VM



Kubernetes



쿠버네티스 난이도

- 쿠버네티스 환경을 구축하는게 더 어려움
- 장애시 원인 파악이 더 어려움
- 생소한 쿠버네티스 개념 이해

비용절감?

리소스 사용 (Cpu, Disk, Memory)
→ 구성하기 나름 (≒ 좋은인재)

오픈소스 사용 (라이선스 비용절감)
→ 오픈소스 관리 인원 투입비용 **증가**

마이크로 서비스 아키텍처에 유리함

- 최소 20개 이상 App
- App 확장 / 상태관리 / 트래픽관리 편리

쿠버네티스 생태계의 빠른 표준화

- 오픈소스들이 어느정도 표준화됨
- 다양한 레퍼런스들이 존재함
- 스스로 공부해서 전문가가 될 수 있는 환경

선진 IT 기업의 이미지

- 쿠버네티스 ≒ IT 인프라의 최신 기술
- 최신 기술을 사용하는 회사를 선호