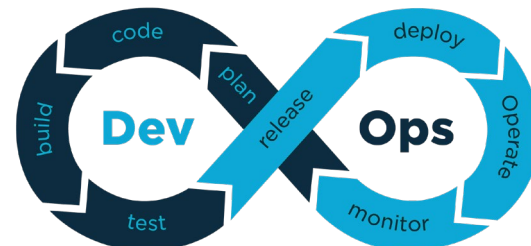
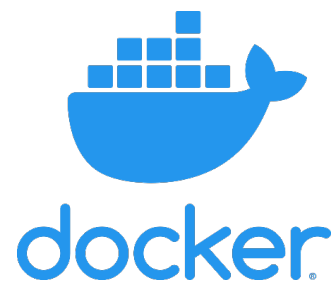
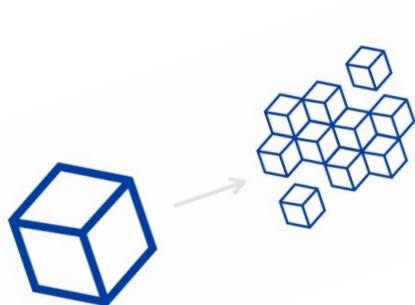


클라우드와 클라우드네이티브

이성미 (SEONGMI LEE)

seongmi.lee@gmail.com



YouTube



파배런





- 클라우드와 클라우드 네이티브
- 클라우드 네이티브 애플리케이션 운영
- Demo: 클라우드 네이티브를 위한 컨테이너 빌드와 운영

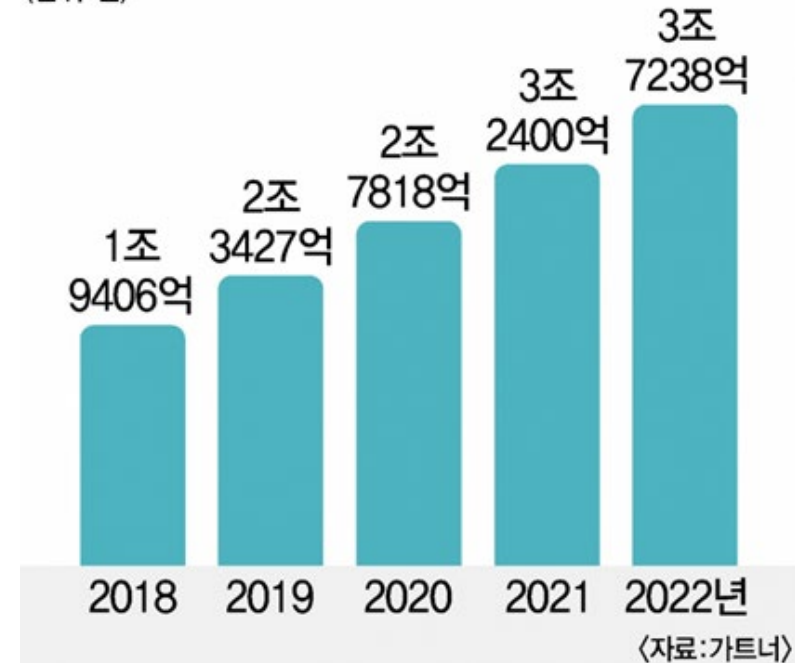
클라우드와 클라우드 네이티브



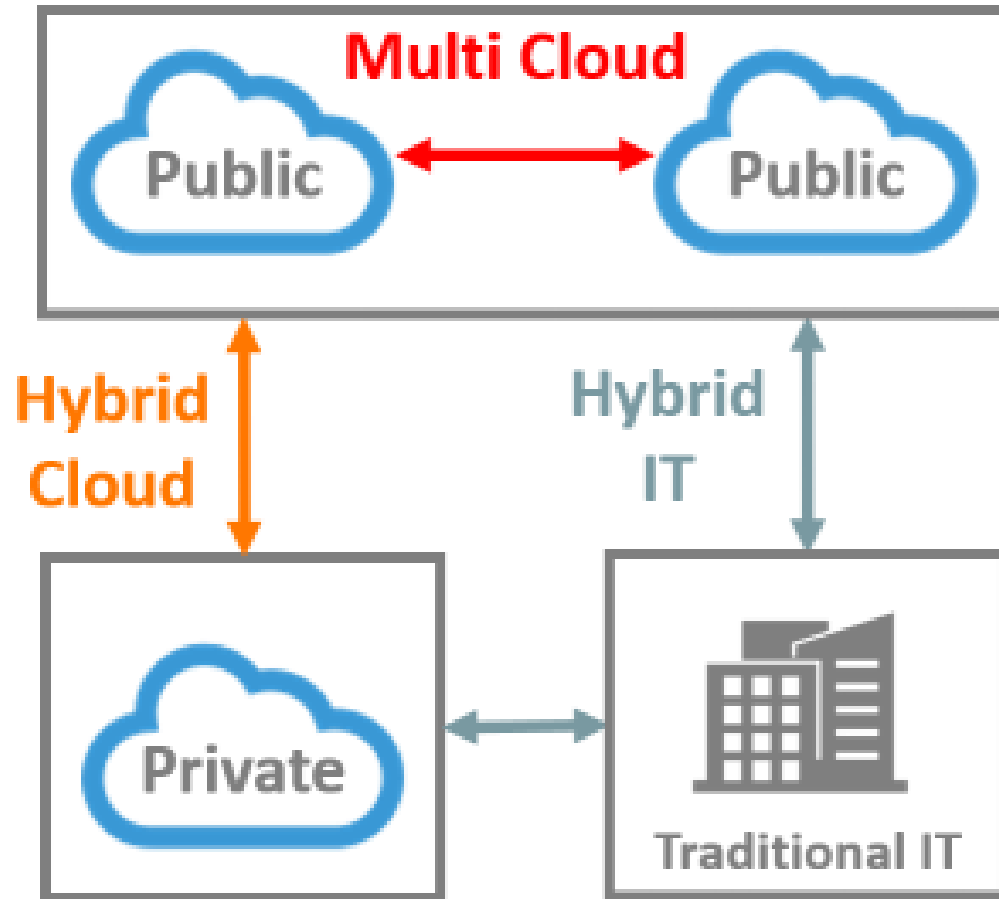
- "컴퓨팅은 언젠가 전화 시스템과 같이 공공재(utility)로서 조직화 될 것입니다. 각각의 가입자들은 실제 그들이 사용한 만큼의 컴퓨팅 파워에 대한 돈을 지불할 것입니다. 컴퓨팅 인프라는 새롭고 중요한 산업의 기반이 될 수 있을 것입니다"
(출처: MIT's centennial celebration, 1961)

- Cloud 산업분야는 COVID19 발생 이후
연평균 16.7% 성장할 것으로 전망

국내 클라우드 시장 전망
(단위: 원)



Private & Hybrid & Multi Cloud



<https://s-bennett.com/cloud-computing/hybrid-multi-cloud-integration-principles/>

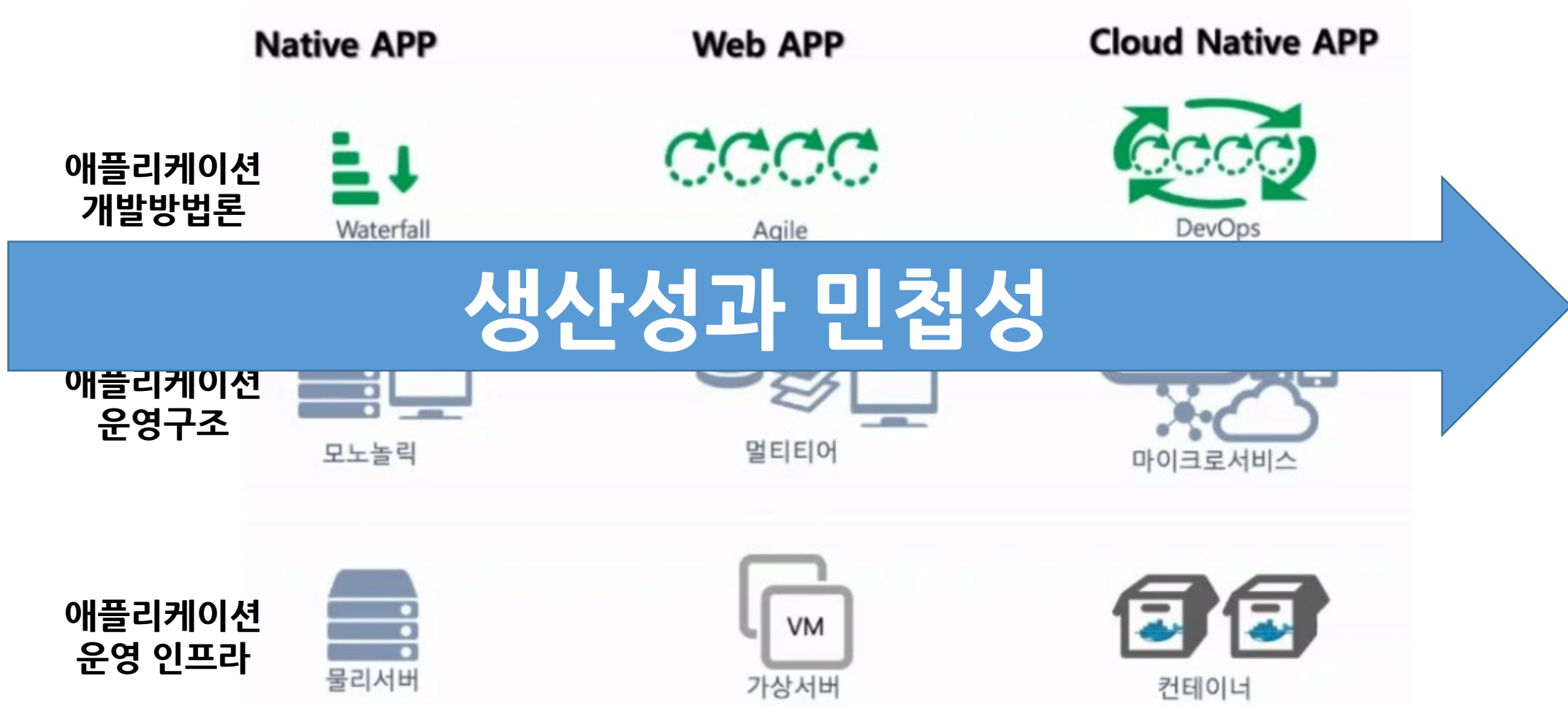


- 클라우드 컴퓨팅 모델을 사용하는 것을 전제로 설계된 시스템과 서비스
 - CNCF: 클라우드 네이티브 기술을 사용하는 조직은 퍼블릭, 프라이빗, 하이브리드 클라우드, 멀티 클라우드와 같이 동적인 환경에서 확장성있는 애플리케이션을 만들고 운영할 수 있다.
 - 컨테이너, 서비스 메시, 마이크로 서비스, 불변의 인프라스트럭처, 그리고 선언적 API가 전형적인 접근 방식에 해당한다.
 - 이 기술은 회복성 있고, 관리 편의성을 제공하며, 가시성을 갖춘 느슨하게 결합된 시스템을 가능하게 한다.
 - 견고한 자동화와 함께 사용하면 엔지니어는 영향이 큰 변경을 최소한의 노력으로 자주, 예측 가능하게 수행할 수 있다.
- 응용 프로그램 개발과 운영이 인프라스트럭처와 운영체제 비 종속적으로 운영

왜 클라우드 네이티브인가?



- 시대적으로 변화되는 애플리케이션 개발환경과 인프라 구조



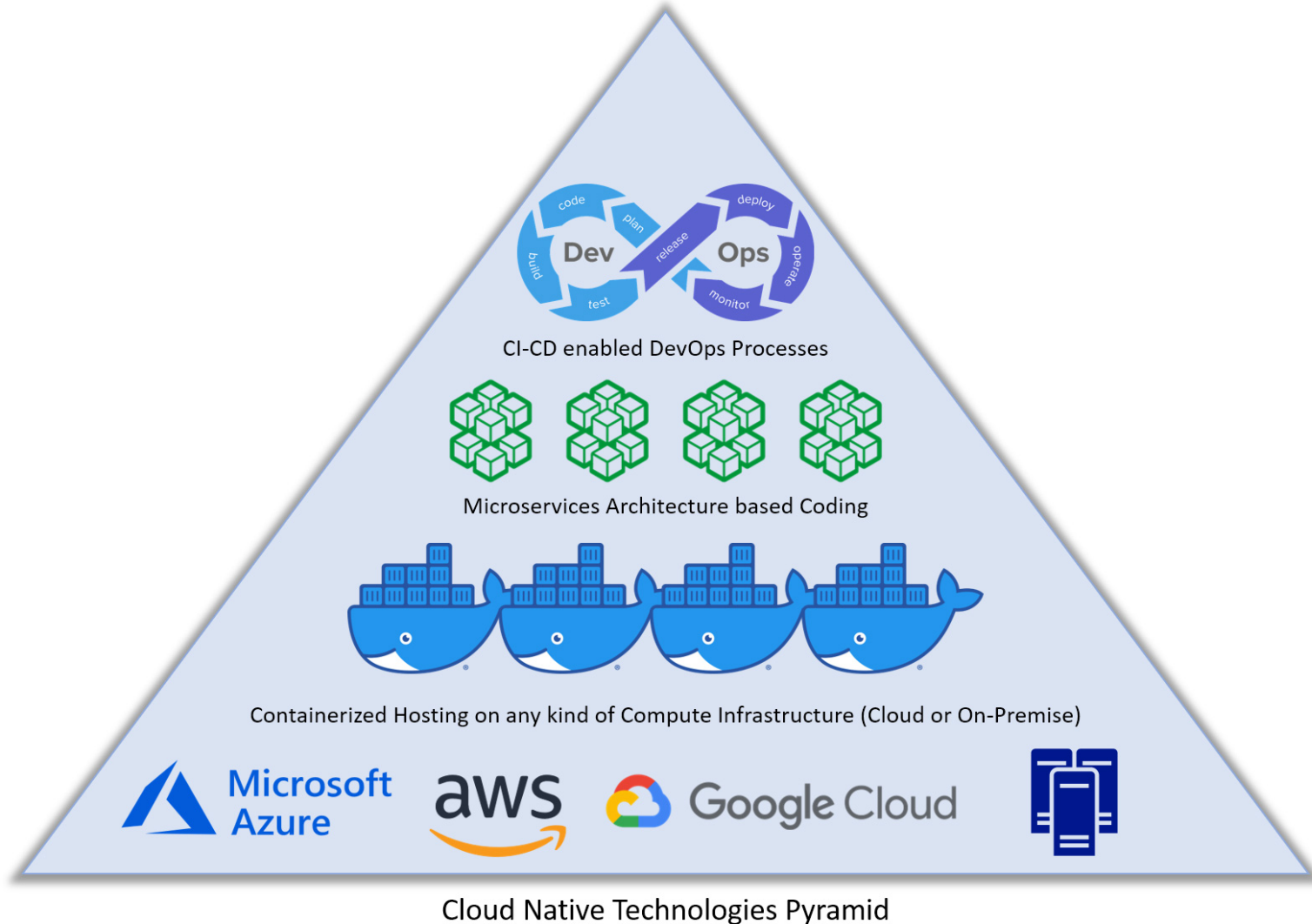
왜 클라우드 네이티브인가?



- 속도
 - 비즈니스의 빠른 변화로 인한 애플리케이션의 잦은 배포 / 신속 배포 / 신속 롤백 필요
- 무중단
 - 투명한 모니터링 / 장애 영향의 최소화 / 자동 복구 체계 필요
 - 애플리케이션 배포, 롤백 시 중단 없는 서비스 요구
- 확장성
 - 요구 사항에 따른 수평적 확장
 - 멀티 클라우드에 대한 투명한 확장 요구

클라우드 네이티브
애플리케이션 운영

Cloud Native 기술 피라미드



Cloud Native 기술 피라미드 : Cloud Platform



- Amazon Elastic Container Service for Kubernetes, Amazon (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)

- Alibaba Cloud
- Oracle Cloud
- Naver Cloud
- KT Cloud



Azure Kubernetes Service (AKS)

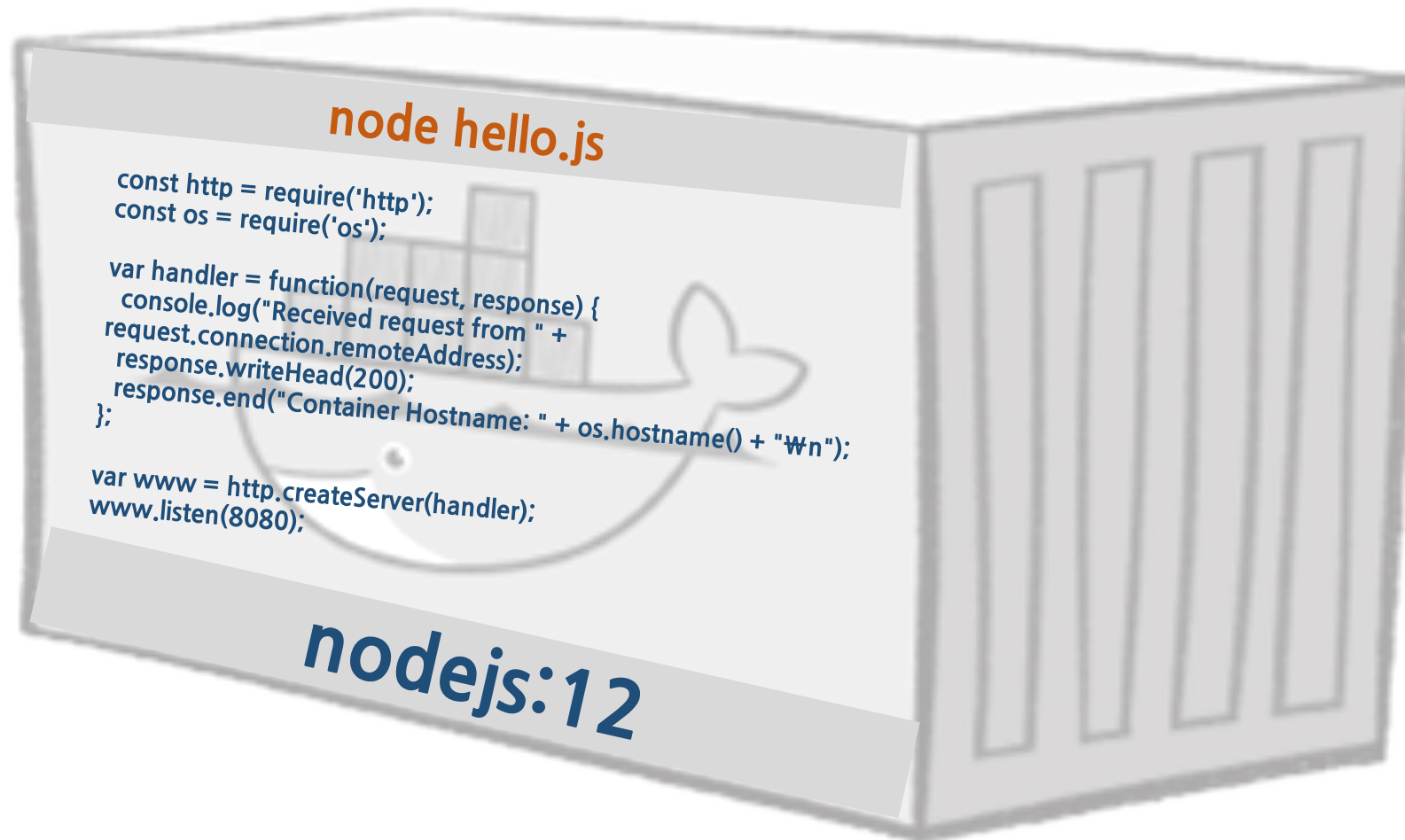


Google Kubernetes Engine





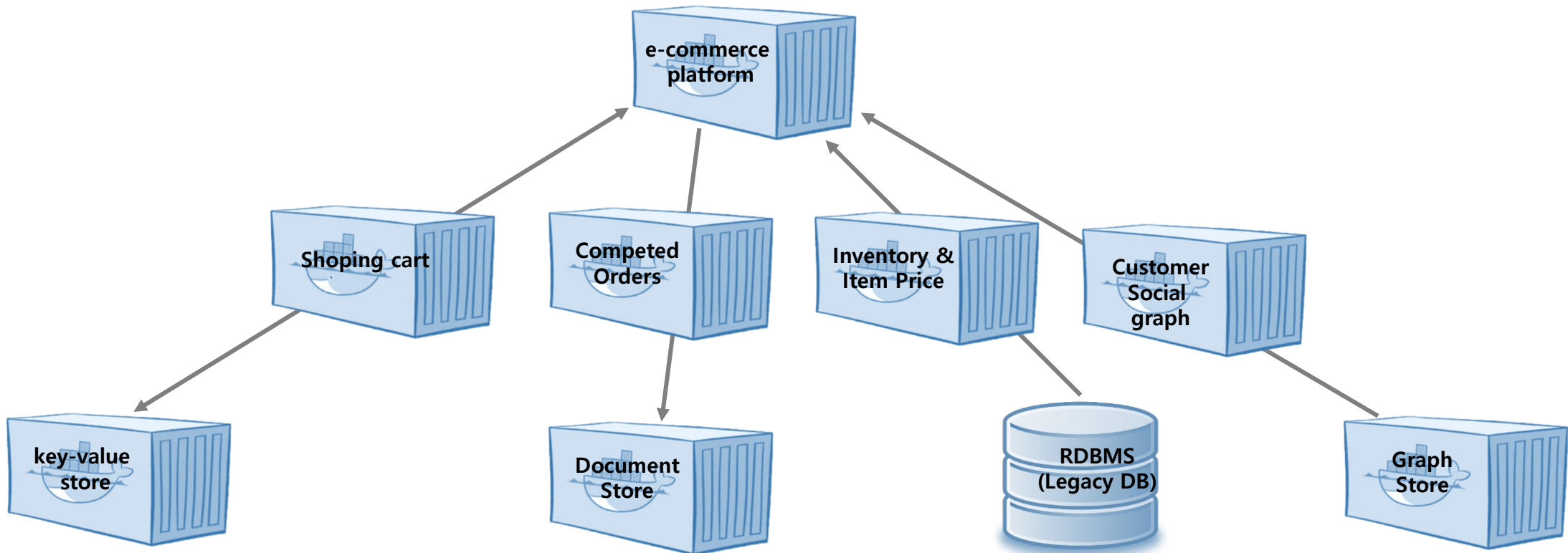
- 개발한 애플리케이션 (실행파일)과 운영환경이 모두 들어있는 독립된 공간



어떻게 컨테이너로 운영할 것인가?



- 개발한 프로그램과 실행환경을 모두 컨테이너로 만들어요
- MSA(Micro Service Architecture) 환경의 Polyglot 애플리케이션 운영





Amazon.com의 사례



아마존은 수 많은 스타트업의 조합이다

수 천개 팀 (자율적 DevOps팀)

× 마이크로서비스 아키텍처

× 지속적 배포 (CD)

× 다양한 개발 환경

= 연간 5천만회 배포

(시간당 5708 회, 또는 0.63 초)

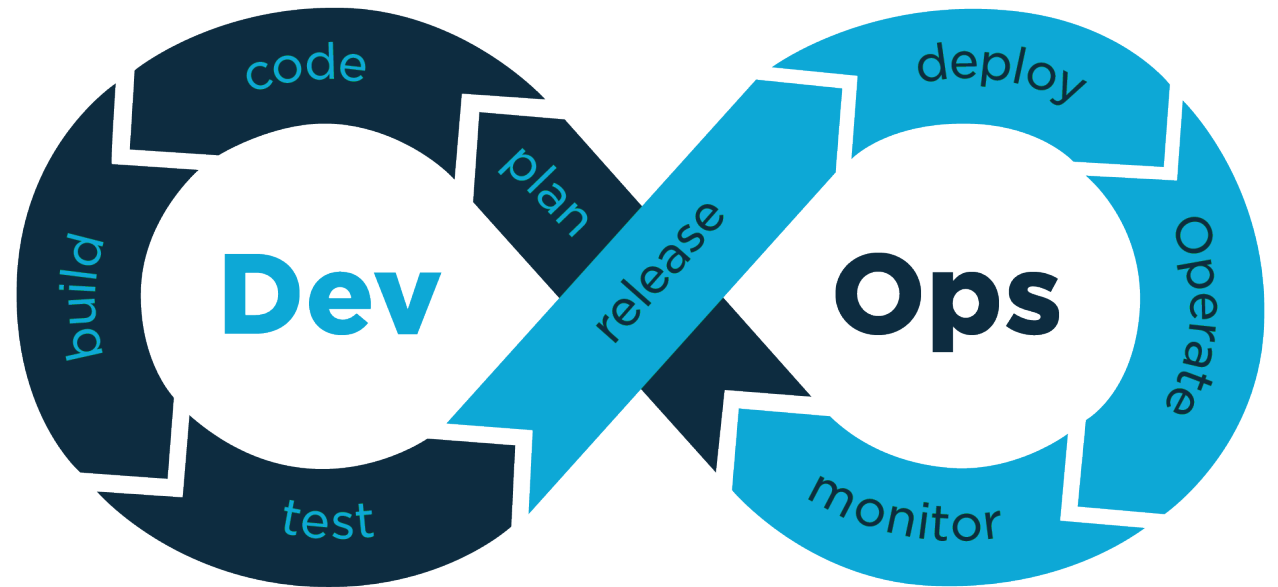
(c) Werner Vogels, The Story of Apollo -
Amazon's Deployment Engine

<http://www.allthingsdistributed.com/2014/11/apollo-amazon-deployment-engine.html>

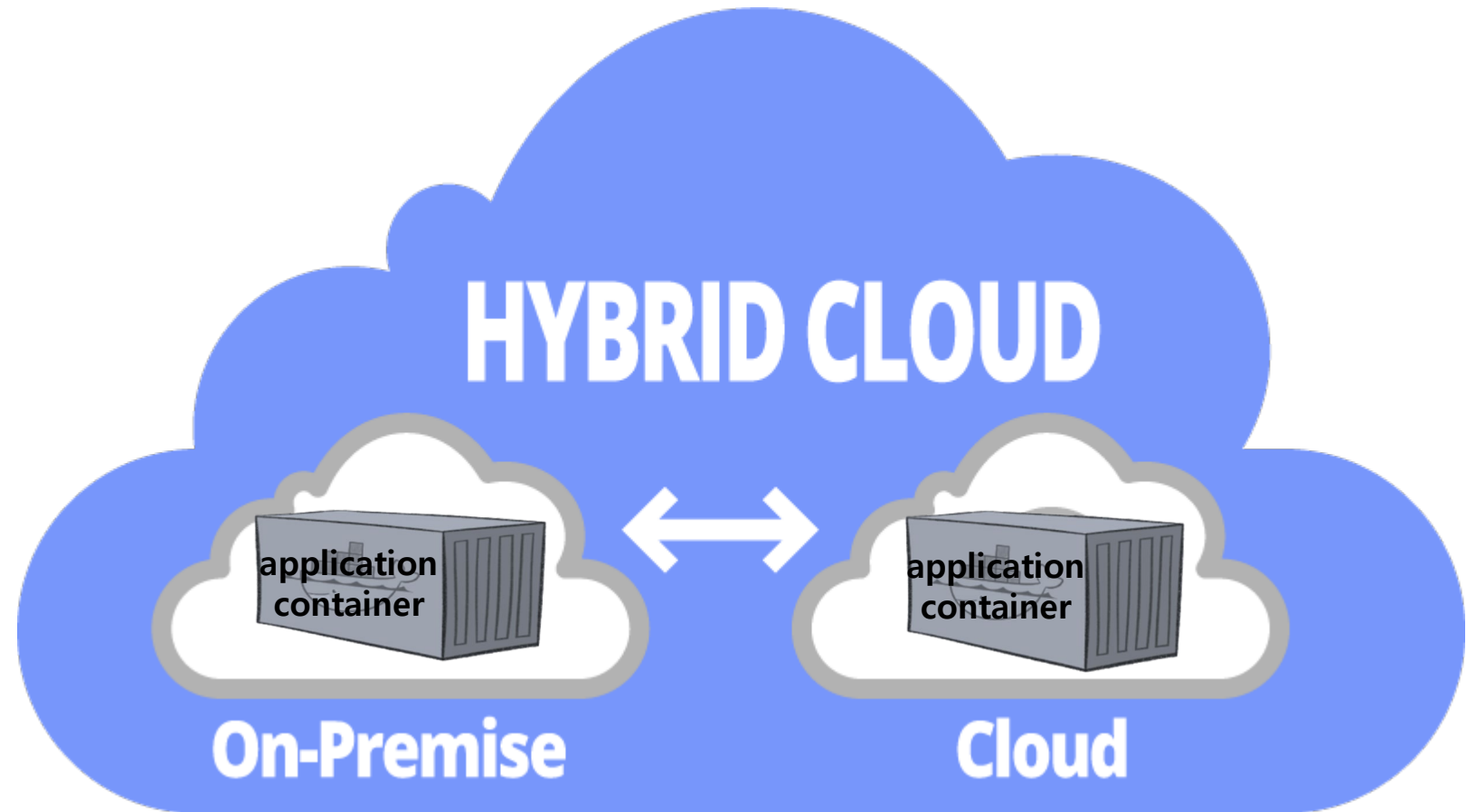
▶ Talk IT



- 개발자와 IT 운영 전문가 간의 소통, 협업 및 통합을 강조하는 개발 운영의 방법론
- SW와 서비스를 빠른 시간에 개발 및 배포하는 것을 목표로 함
 - 소스
 - 코드리뷰
 - 빌드
 - 배포
 - 이슈관리
 - 지속적인 파이프라인의 프로세스 정립
- 자동화된 라이프사이클

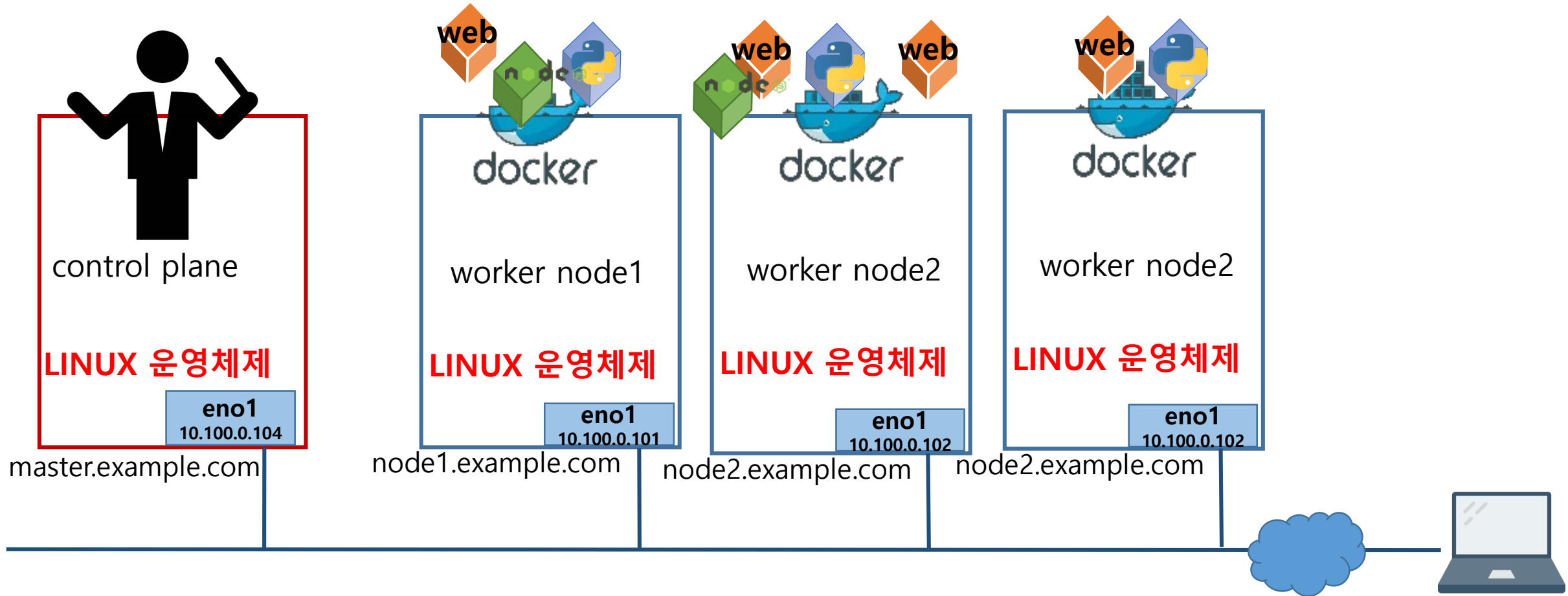


Cloud Native 기술 피라미드 : PaaS 플랫폼 확장성





컨테이너 오케스트레이션





Layer 6	Development Workflow Opinionated Containers	OpenShift, Cloud Foundry, Docker Cloud, Deis, Flynn ...
Layer 5	Orchestration/Scheduling Service Model	Kubernetes, Docker Swarm, Marathon/Mesos, Nomad, Diego ..
Layer 4	Container Engine	Docker, Rocket, RunC(OCI), Osv, LXC, LXD
Layer 3	Operating System	Ubuntu, RHEL, CoreOS, Unikernels
Layer 2	Virtual Infrastructure	vSphere, EC2, GCP, Azure, OpenStack
Layer 1	Physical Infrastructure	Raw Computer, Network, Storage

DEMO:

클라우드 네이티브를 위한

컨테이너 빌드와 운영

Demo: 클라우드 네이티브를 위한 컨테이너 빌드와 운영



- 웹 서비스를 위한 컨테이너 빌드
- 컨테이너 운영
- Kubernetes 오케스트레이션을 이용해 서비스 운영
- 스케일링

Kubernetes Application Service



쿠버야 payment 앱을 3개 실행해줘!

쿠버야 marvel-collection 파드1개 실행해 줘

