

IT 생존을 위한 진정한 클라우드 기술!
이젠 클라우드 네이티브를 꼭 알아야 합니다.

Development Process



WATERFALL



AGILE



DEVOPS



Application Architecture



MONOLITHIC



N-TIER



MICROSERVICES



Deployment & Packaging



PHYSICAL SERVERS



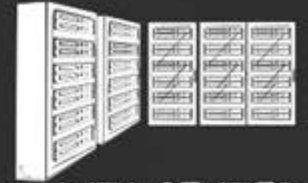
VIRTUAL SERVERS



CONTAINERS



Application Infrastructure



DATA CENTER



HOSTED



CLOUD



클라우드 네이티브를 저해하는 요인

- 생각과 시스템을 **클라우드 네이티브** 하게 전환하지 못하면, 클라우드라도 개선이 없음



클라우드를 임대하여 사용하는 것일 뿐

- 가상머신과 스토리지를 임대하여 사용하고 있을 뿐, 기존의 인프라와 다르지 않음



클라우드 특징에 맞게 설계하고 운영 하지 않음

- 클라우드 특성을 이해하지 못하고 기존 인프라를 단순히 대체하여 설계
- 비용 부분에서만 정액제 클라우드로 전환하였으나, 벤더 종속성과 비용만 높아짐



인프라만 클라우드 일 뿐 조직은 그대로

- 기존의 개발팀과 운영팀이 수행하던 역할과 프로세스 그대로 운영



클라우드로 전환했으나 구인난과 고비용 구조로 더 큰 문제

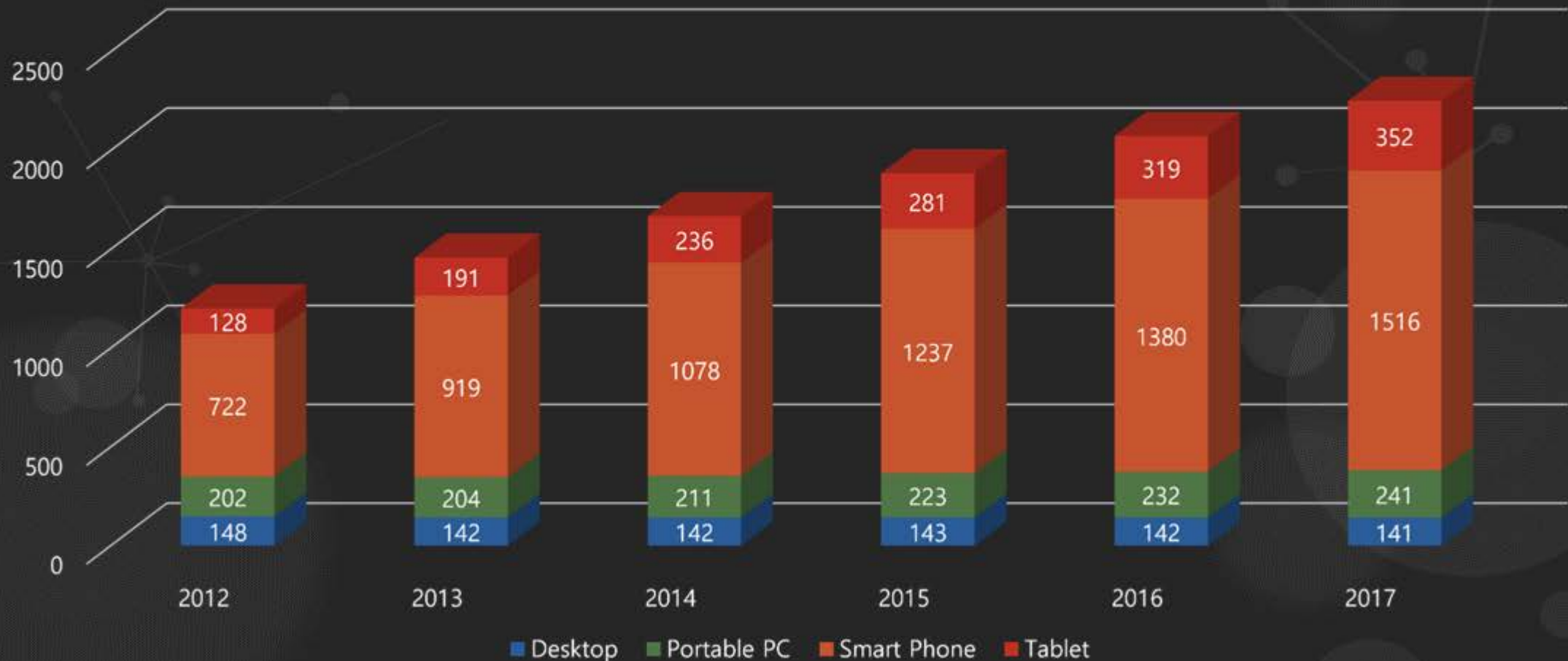
- 클라우드를 이용하고 있음에도 불구하고 수작업 프로세스에서 벗어나지 못함
- 운영 인력 부족과 업무 효율성을 개선하지 못함



Digital Transformation

Smartphone Shipments to Top 1 Billion in 2014

- Global connected device shipment forecast (in million units)



Tipping point in the digital revolution

Infrastructure as Code & DevOps Seminar



Tipping point in the digital revolution

엑스 4층
룸 402호
수) 13:00~18:00



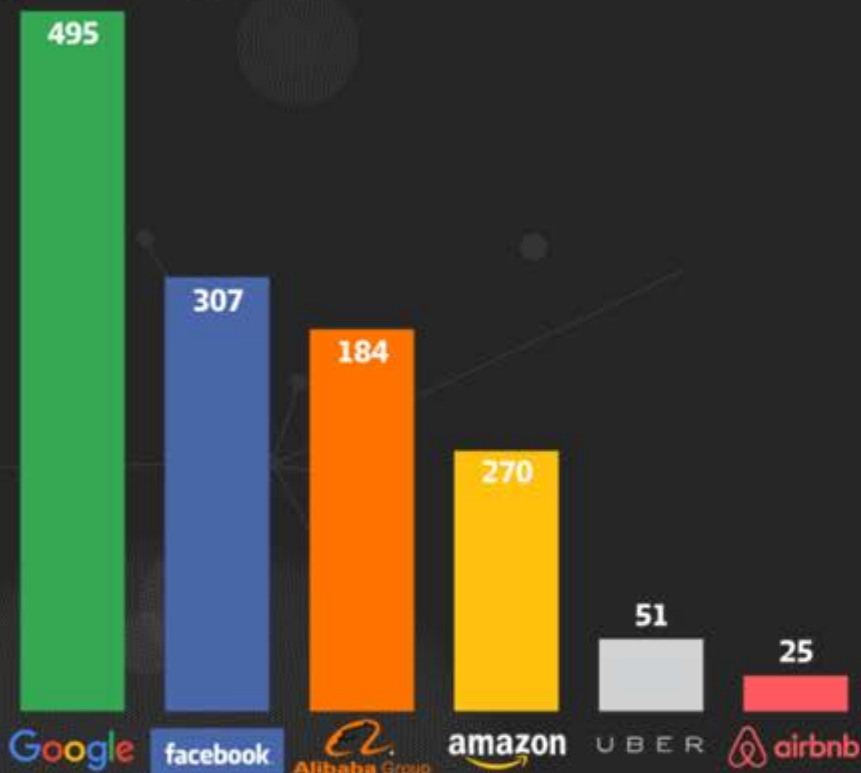
파괴적 혁신 시대의 도래



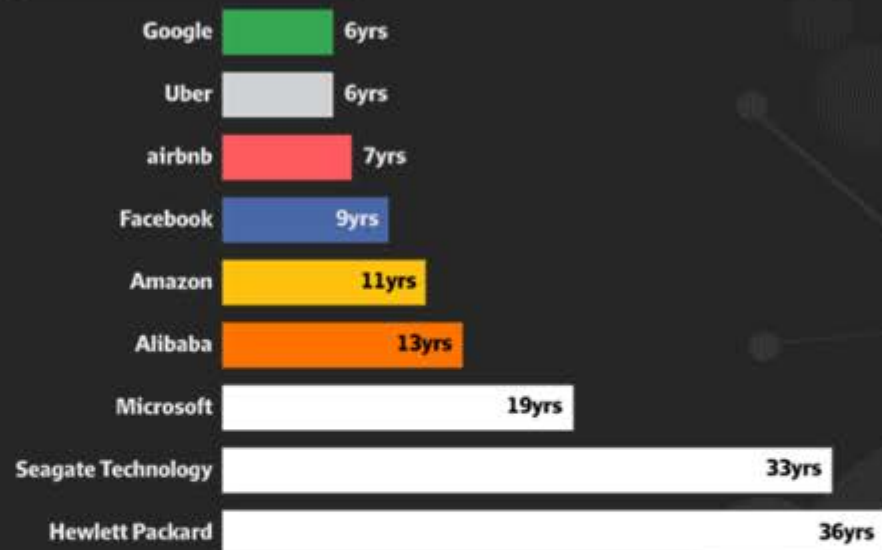
디스래퍼터	디스러프트된 산업
Apple	음악 산업
Google	광고 산업
Uber	택시 산업
Airbnb	호텔 산업
아마존	소매업계

Digital Disruption: The new giants are different

Valuation (\$USb)



Years taken to join the S&P 500



Founders



<http://www.afr.com/news/special-reports/transformation-agenda/google-facebook-alibaba-amazon-uber-and-airbnb-show-older-businesses-how-its-done-20160315-gnk0xb>

Business Disruptor



- 자신의 방이나 집, 별장 등 사람이 지낼 수 있는 모든 공간을 임대할 수 있으며, 192개국 3만 4800여 개 장소에서 200만여 개의 객실에 대한 숙박을 중개
- Airbnb는 부동산을 소유하지 않음



- 전 세계 14억 9천 만명 이상의 월 활동 사용자가 활동 중인 세계 최대의 소셜 네트워크 서비스
- 자체 생산 콘텐츠 없이도 뉴스, 사진, 비디오 제공



- 승객이 스마트폰 앱을 이용해 차량을 호출하면 우버와 계약한 기사가 자기 차량을 몰고와 목적지까지 데려다 주는 '주문형 개인 기사 서비스'
- Uber는 보유한 차량이 없음



- 중국의 전자상거래업체로 '세계에서 가장 큰 쇼핑몰' 서비스
- Alibaba는 보유한 상품재고 없이 전자상거래

ALL CHECKED IN!

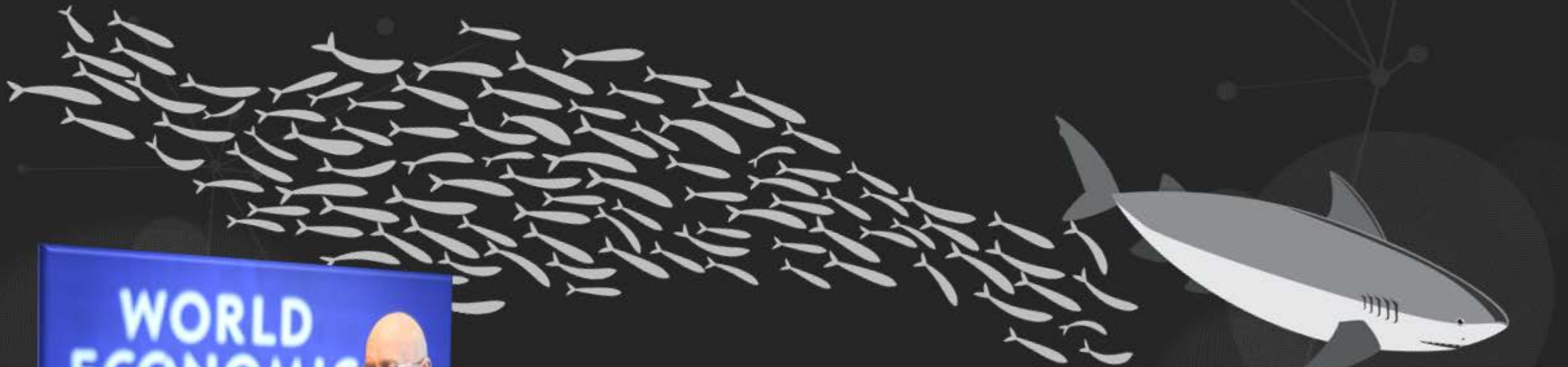


ONLINE STORE



The fast fish eats the slow fish

“In the new world, it is not the **big fish** which eats the **small fish**,
it’s the **fast fish** which eats the **slow fish**.”



4 차 산업혁명의 이해 (Mastering the Fourth Industrial Revolution) – 세계 경제 포럼
Klaus Schwab, Founder and Executive Chairman of the World Economic Forum

소프트웨어 시대

- 소프트웨어 개발의 혁신
 - 적은 인원으로 대규모 개발이 가능해짐
 - 오픈소스 소프트웨어 활용
 - 자동화 도구 발전
 - 클라우드 컴퓨팅 등장
 - DevOps와 Agile개발의 등장
 - 빠른 구현에서 배포에 이르는 사이클을 안정적 컨트롤 하는 기술과 프로세스가 등장
- 기업에 있어서 IT 위상변화
 - 기업 혁신을 위한 핵심 역량으로 IT 가 대두
 - AI/ML , 자동화,
 - 산업의 고도화로 인해 점차 핵심분야로 이동
 - 핵심비즈니스 영역에 포함되거나 아예 핵심 비즈니스 자체가 됨



“**메르세데스 벤츠**는 이제 자동차 기업이 아닌 소프트웨어 기업이며 자동차는 궁극의 웨어러블이다.

- 올라켈레니우스, 메르세데스 벤츠 그룹 대표



“우리는 클라우드 네이티브로 기술을 재구축하면서 조직 운영 방식도 바꿨다.

- 넷플릭스 클라우드 부문 부사장 이즈라일레브스키



“**골드만삭스**는 **정보기술(IT)** 회사다.

- 골드만삭스 CEO 데이비드 리브킨



“인간 중심의 **디지털 전략**이 미래 **리테일 산업**의 **성공의 열쇠**가 될 것입니다.

- Kevin Johnson, CEO of Starbucks



SoR 에서 SoE 로 전환

Systems of Record (SoR)

	Mainframe	Mini	PC
Time Frame	1960 - 1975	1975-1992	1992-2001
Data Types	Batch	Dept Process	Documents
First Movers	IBM, Unisys	DEC, Compaq	Microsoft, Dell, IBM

Systems of Engagement (SoE)

	Internet	Mobile/Cloud	Connected
Time Frame	2001-2009	2010-2015	2016-
Data Types	Web Pages	User Interactions	IOT / AI
First Movers	Google, Microsoft	Facebook, Amazon, Apple	Airbnb, Uber

데이터를 기록하는 시스템
신뢰성/안전성/정확성/데이터 중심
Waterfall
Java & .NET
독점 기술 중심/ 물리&가상 환경
모노리식
웹기반
일괄/실시간 처리



연결·관계·상호작용을 만드는 시스템
즉시성/유연성/사용자 중심
DevOps & Agile
경량/다양한 프로그램 언어
오픈 / 클라우드
마이크로 서비스
Mobile
리얼타임, 탄력적인 인프라

기존 IT로부터 새로운 IT로의 과도기

기존 IT



- 물리 서버와 PC
- 5년 이상 시스템 사용 기간
- 폭포수형 개발
- 운용의 우선 사항 "안정성"
- 시스템 부문은 시스템에 전념
- 대형 IT 벤더가 기술을 주도



새로운 IT



- 클라우드와 모바일
- 몇 개월 사용 후 폐기
- Agile, DevOps
- 운용의 우선 사항 "민첩성"
- 시스템 부문도 비즈니스에 초점
- 사용자 주도 기술, **오픈 소스 주도**

디지털 트랜스포메이션 정의



구분	정의
Bain & Company	디지털 엔터프라이즈 산업을 디지털 기반으로 재정의하고 게임의 법칙을 근본적으로 뒤집음으로써 변화를 일으키는 것임
AT Kearney	모바일, 클라우드, 빅데이터, 인공지능 (AI), 사물인터넷 (IoT) 등 디지털 신기술로 촉발되는 경영 환경상의 변화에 선제적으로 대응하고 현재 비즈니스의 경쟁력을 획기적으로 높이거나 새로운 비즈니스를 통한 신규 성장을 추구하는 기업 활동임
PWC	기업경영에서 디지털 소비자 및 에코시스템이 기대하는 것들을 비즈니스 모델 및 운영에 적용시키는 일련의 과정임
Microsoft	고객을 위한 새로운 가치를 창출하기 위해 지능형 시스템을 통해 기존의 비즈니스 모델을 새롭게 구상하고 사람과 데이터, 프로세스를 결합하는 새로운 방안을 수용하는 것임
IBM	기업이 디지털과 물리적인 요소들을 통하여 비즈니스 모델을 변화 (Transform) 시키고 산업 (Entire Industries) 에 새로운 방향 (New Directions) 을 정립하는 것임
IDC	고객 및 마켓 (외부환경) 의 변화에 따라 디지털 능력을 기반으로 새로운 비즈니스 모델, 제품 서비스를 만들어 경영에 적용하고 주도하여 지속 가능하게 만드는 것임
World Economic Forum	디지털 기술 및 성과를 향상시킬 수 있는 비즈니스 모델을 활용하여 조직을 변화시키는 것임

Source - <https://digitaltransformation.co.kr/>

Are you a digital transformer?



- 디지털 세계에서 점점 더 많은 비즈니스가 발생함에 따라 고객 서비스에 중점을 둔 신속하고 데이터 기반 의사 결정을 내리는 것이 중요합니다.
- 기술은 올바르게 사용된다면 비즈니스 성장에 도움이 되는 훌륭한 도구가 될 수 있습니다.
- 디지털 트랜스포머는 비즈니스에 대한 확고한 이해와 기술을 사용하여 비즈니스 성장을 돕는 의사 결정 프로세스를 주도하는 방법을 알고 있는 사람입니다.

Are you a digital follower or a digital transformer?

Cloud & Open Source

Emerging Cloud Computing Technologies

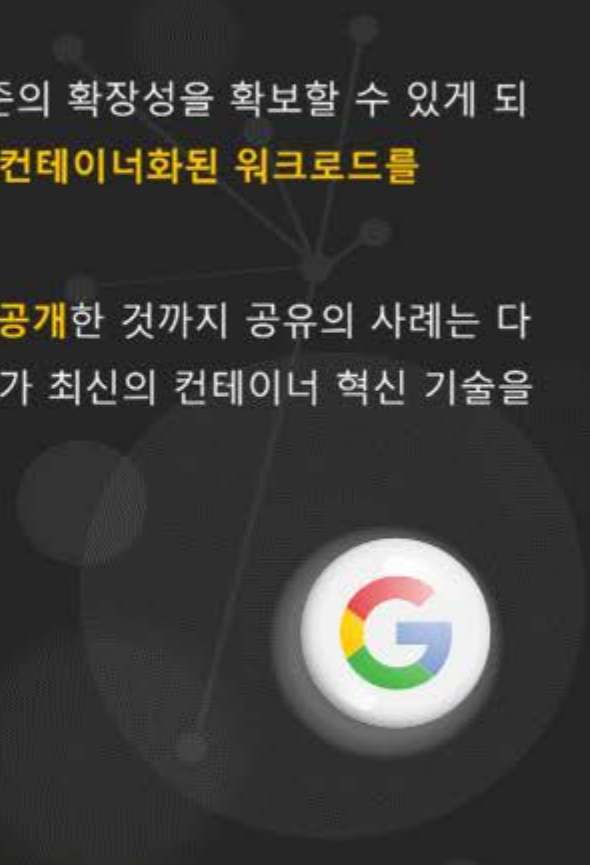
GOOGLE 과 컨테이너

- Google의 업무 방식

Gmail에서 YouTube, 검색에 이르기까지 Google의 모든 제품은 컨테이너에서 실행됩니다.

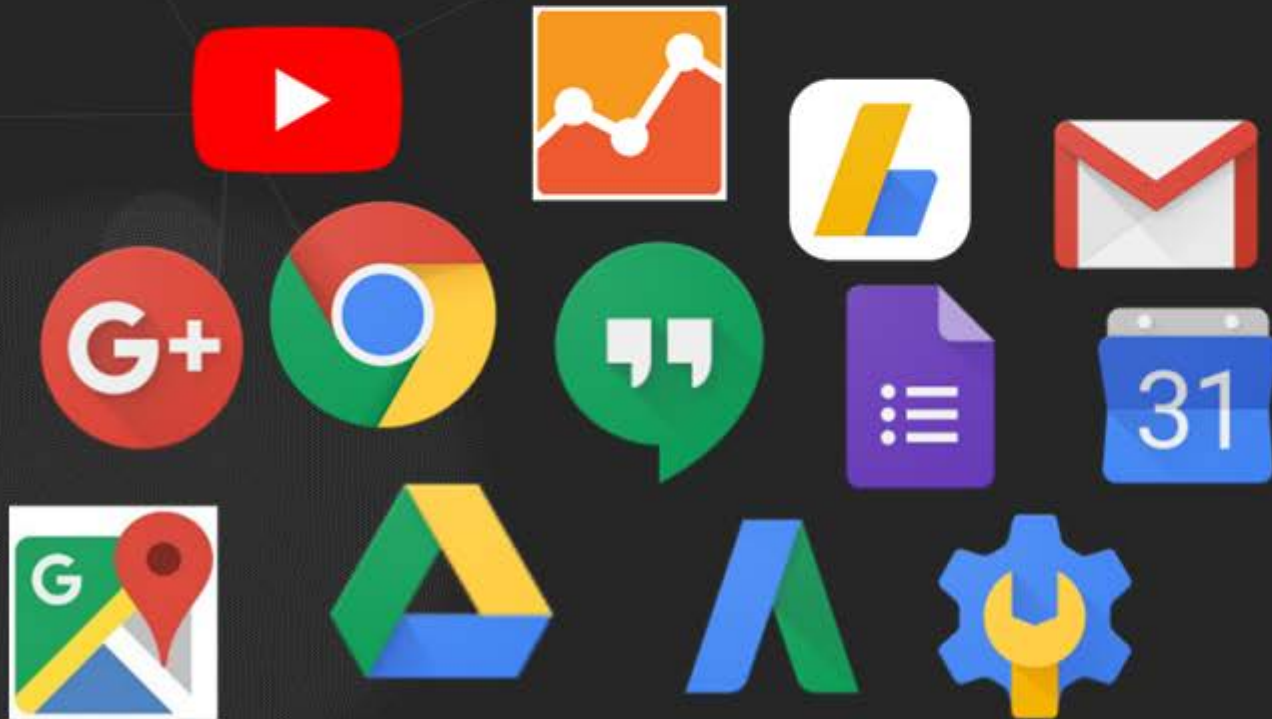
개발팀은 컨테이너화를 통해 더욱 신속하게 움직이고, 효율적으로 소프트웨어를 배포하며 전례 없는 수준의 확장성을 확보할 수 있게 되었습니다. Google은 매주 수십억 개가 넘는 컨테이너를 생성합니다. 지난 10여 년간 프로덕션 환경에서 컨테이너화된 워크로드를 실행하는 방법에 관해 많은 경험을 쌓으면서 Google은 커뮤니티에 계속 이 지식을 공유해 왔습니다.

초창기에 cgroup 기능을 Linux 커널에 제공한 것부터 내부 도구의 설계 소스를 Kubernetes 프로젝트로 공개한 것까지 공유의 사례는 다양합니다. 그리고 이 전문 지식을 Google Cloud Platform으로 구현하여 개발자와 크고 작은 규모의 회사가 최신의 컨테이너 혁신 기술을 쉽게 활용할 수 있도록 하였습니다.



Google 는 모두 컨테이너에서 실행

- Gmail , 검색, 지도 ...
- MapReduce , GFS , Colossus ...
- Google Compute Engine 가상 머신도 컨테이너에서 실행!
- 매주 20 억개 이상의 컨테이너를 실행 중



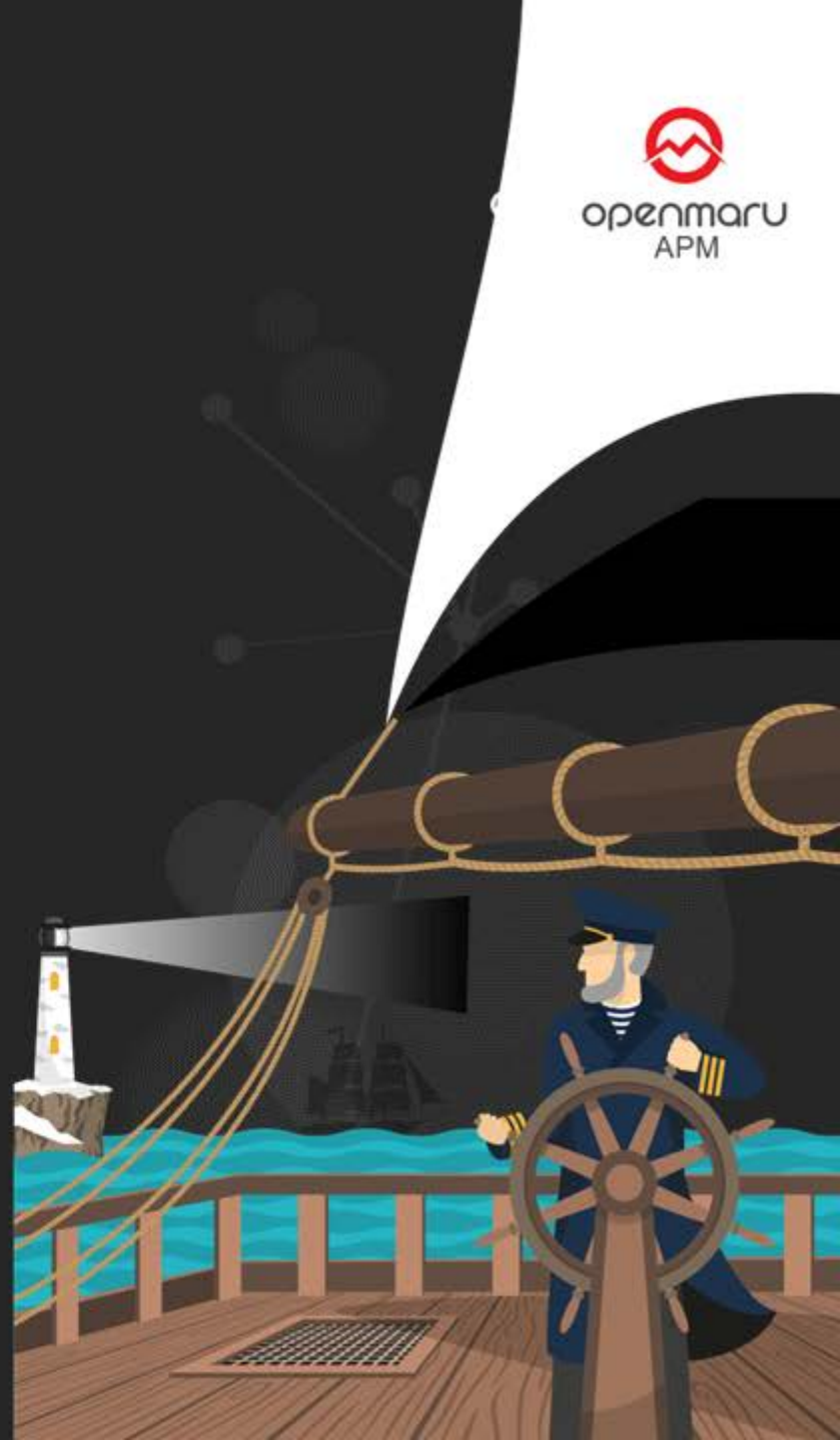
About Kubernetes

- 쿠버네티스(K8s)는 컨테이너화된 애플리케이션을 자동으로 배포, 스케일링 및 관리해주는 오픈소스 소프트웨어
- 쿠버네티스", "쿠베르네티스", "K8s", "쿠베", "쿠버", "큐브"라고 부르며 Apache License 2.0 라이선스로 리눅스 재단 (Linux Foundation)산하 Cloud Native Computing Foundation (CNCF) 에서 관리
- Go로 작성된 오픈 소스 , OSS (Apache License 2.0) 라이선스
- 구글에서 개발하고 설계한 플랫폼으로서 사내에서 이용하던 컨테이너 클러스터 관리 도구인 "Borg"의 아이디어를 바탕으로 개발

"Kubernetes is open source-a contrast to Borg and Omega, which were developed as purely Google-internal systems. "

- Borg, Omega, and Kubernetes

- Confidential -



컨테이너는 클라우드 에서 Java 와 같이 벤더 종속성 해제

2000 년 - Java 를 통한 Vendor Lock-In 해제



2020 년 - 컨테이너와 Kubernetes 를 통한 Vendor Lock-In 해제



CNCF Cloud Native Definition v1.0



클라우드 네이티브 기술을 사용하는 조직은 현대적인 퍼블릭, 프라이빗, 그리고 하이브리드 클라우드와 같이 동적인 환경에서 확장성 있는 애플리케이션을 만들고 운영할 수 있다.

컨테이너, 서비스 메시, 마이크로서비스, 불변의 인프라스트럭처, 그리고 선언적 API가 전형적인 접근 방식에 해당한다.

이 기술은 회복성이 있고, 관리 편의성을 제공하며, 가시성을 갖는 느슨하게 결합된 시스템을 가능하게 한다.

견고한 자동화와 함께 사용하면, 엔지니어는 영향이 큰 변경을 최소한의 노력으로 자주, 예측 가능하게 수행할 수 있다.

Cloud Native Computing Foundation은 **벤더 중립적인 오픈소스 프로젝트 생태계**를 육성하고 유지함으로써 해당 패러다임 채택을 촉진한다.

우리 재단은 최신 기술 수준의 패턴을 대중화하여 이런 혁신을 누구나 접근 가능하도록 한다.

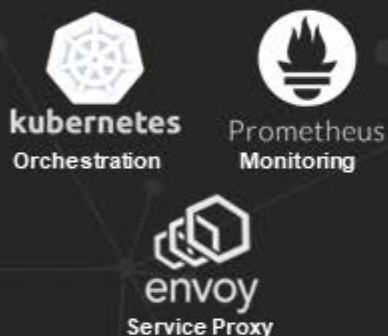


Cloud Native Computing Foundation

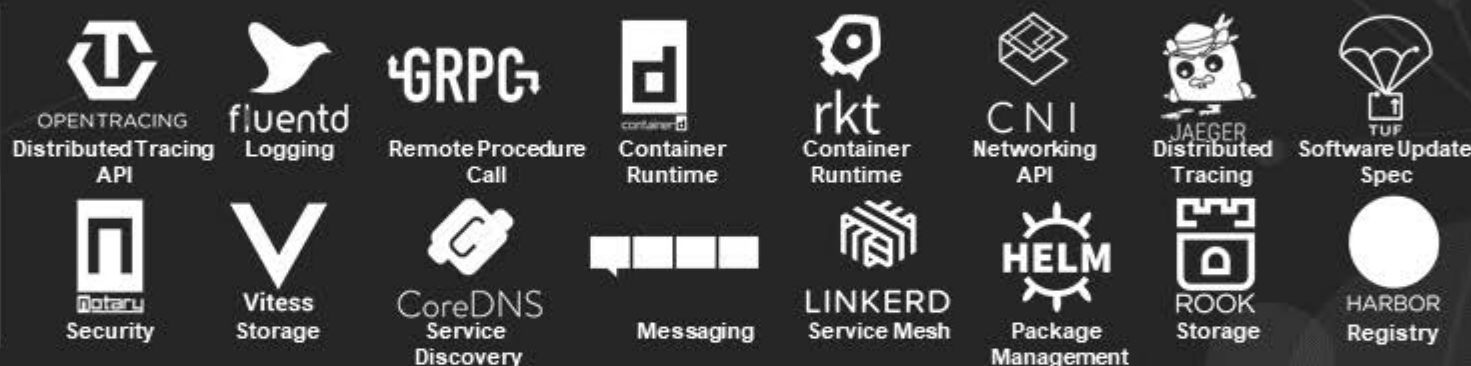


- Non-profit, part of the Linux Foundation; founded Dec 2015

Graduated



Incubating



- Platinum members:





“누군가가 나의 등잔의
심지에서 불을 붙여가도
내 등잔의 불은 여전히
빛나고 있습니다.”

미국의 정치가 토머스 제퍼슨

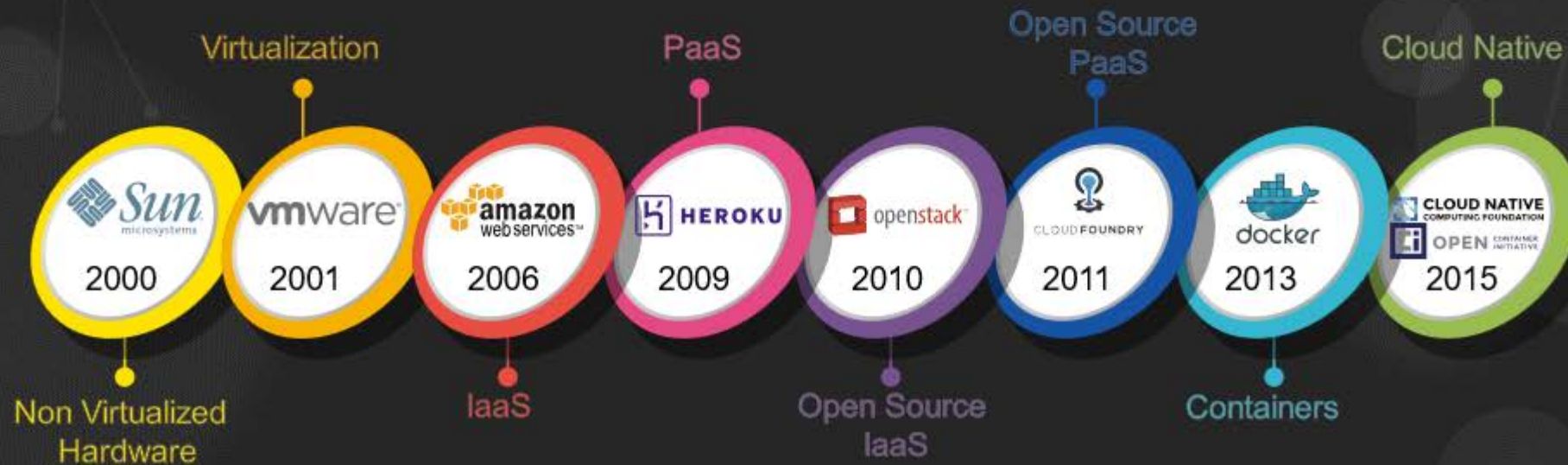
Cloud Native



Cloud Native vs. Cloud Immigrant

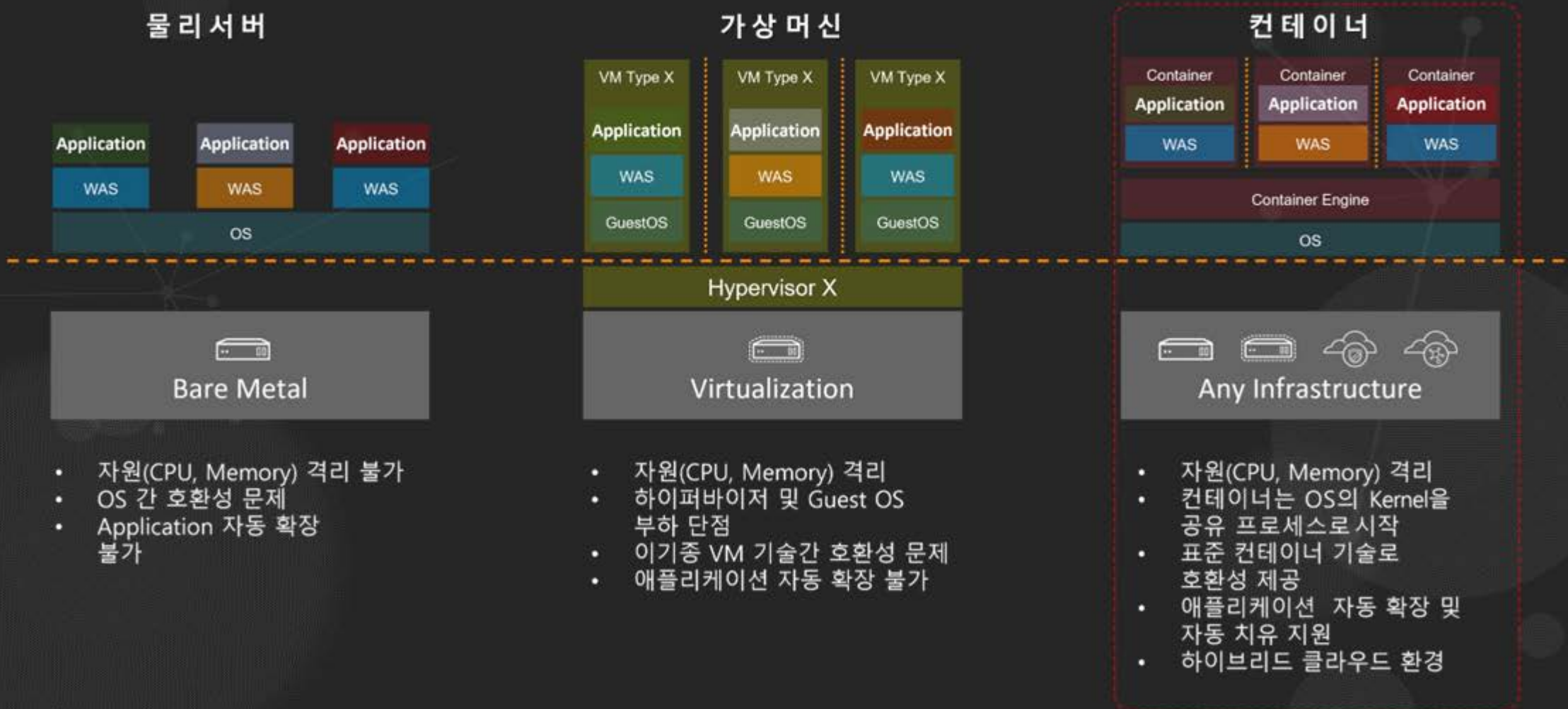
From Virtualization to Cloud Native

- Cloud native computing uses an open source software
- Standardization : <https://www.opencontainers.org/>



WHY CONTAINER ?

- 자원 효율성, 자원 격리, 호환성, Auto Scaling, DevOps, MSA, 관리 편의성



- 자원(CPU, Memory) 격리 불가
- OS 간 호환성 문제
- Application 자동 확장 불가

- 자원(CPU, Memory) 격리
- 하이퍼바이저 및 Guest OS 부하 단점
- 이기종 VM 기술간 호환성 문제
- 애플리케이션 자동 확장 불가

- 자원(CPU, Memory) 격리
- 컨테이너는 OS의 Kernel을 공유 프로세스로 시작
- 표준 컨테이너 기술로 호환성 제공
- 애플리케이션 자동 확장 및 자동 치유 지원
- 하이브리드 클라우드 환경

Cloud Immigrant vs. Cloud Native



구분	Cloud Immigrant	Cloud Native
서비스 모델	가상화 기반 IaaS (Infrastructure As A Service)	컨테이너 기반 PaaS (Platform As A Service)
디자인	On Premise 에 구축된 시스템을 클라우드로 이전하여 운영	시작 단계부터 클라우드의 장점인 민첩성, 확장성 그리고 이동성을 최대한 활용할 수 있도록 설계
구현	특정 클라우드 벤더에 의존적인 설정이 있어 구축에 시간이 걸림	어떤 클라우드 환경에서도 빠르고 효율적으로 전환 (Portability)
확장성	애플리케이션 업데이트가 수작업이기 때문에 장시간의 다운타임일 필요하고 Scale In/Out 이 어려움	컨테이너와 MSA 기반으로 서비스에 영향을 주지 않고, 업데이트가 필요한 서비스만 변경할 수 있으며, 서비스 단위의 Scale In/out 지원
비용	애플리케이션이 커질 수록 인프라 비용이 상승	인프라 부분의 종속성이 없어 비용이 저렴
유지보수	버전관리, 설치 그리고 구성관리가 수작업이며 복잡함	CI (Continuous Integration) / CD (Continuous Delivery)

물리서버 시대 ~ 2000년

- 모놀리스 애플리케이션 운영
- 물리서버 댓수가 많지 않고, 서버를 1:1로 관리



~2000년
물리서버

2001-2009년
가상화 기술 1세대

2010-2015년
가상화 기술 2세대

2016년 -
클라우드 네이티브

가상화 기술 1세대 : 2001년 ~ 2009년



- 물리서버를 가상머신으로 대체
- 통합 비율을 올리고 고효율화 하는 것이 목적
- 서버의 멀티 코어 화와 가상화 기술의 보급

2001~2009년
가상화 기술 1세대

~2000년
물리서버

2010~2015년
가상화 기술 2세대

2016년 ~
클라우드 네이티브

가상화 기술 2세대 ≙ Cloud 시대: 2010년 ~ 2015년

- 클라우드가 대규모 서비스에 적합하고 확장 가능성을 입증
- 안정적으로 대규모 인프라를 관리하기 위한 기술도 보급
 - Immutable Infrastructure/ Infrastructure as Code



IBM Cloud



Google Cloud



Cloud Native 시대 2016 년 ~

- 2016 년 1 월에 정식 출범 한 Cloud Native Computing Foundation (이하 CNCF)는 진짜 클라우드 기술을 오픈소스로 해결하는 하는 것을 목표
- 애플리케이션을 실행하는 데 필요한 최적의 인프라 제공
 - 개발 한 것을 "즉시" "안정적으로" 제공 (비즈니스 우위)



Google Cloud



kubernetes

2016년 ~
클라우드 네이티브

~2000년
물리서버

2001-2009년
가상화 기술 1세대

2010-2015년
가상화 기술 2세대

Cloud Native ERA

- 애플리케이션을 실행하기 위한 최적의 인프라 최적 솔루션 중 하나
 - 컨테이너 기술
 - 애플리케이션이 동작하기 위한 운영 환경을 함께 패키징
 - 개발자를 위한 이미지 빌드/배포 용이성
 - 빠른 애플리케이션 실행과 낮은 오버헤드
 - Kubernetes(컨테이너 오케스트레이션)
 - 애플리케이션 실행을 먼저 생각할 때
 - 어떤 인프라를 만들지 주축으로 설계된 인프라 기반

Developer Experience 장점

Reconciliation model의 정교함

2016년 ~
클라우드 네이티브

~2000년
물리서버

2001~2009년
가상화 기술 1세대

2010~2015년
가상화 기술 2세대

Cloud Native Computing으로 전환 효과

- Cloud Native Computing 환경은 클라우드가 제공하는 민첩성, 가용성, 확장성의 장점을 어플리케이션/서비스의 개발, 운영, 관리에 적용하여 기존 컴퓨팅 환경을 최적화 함



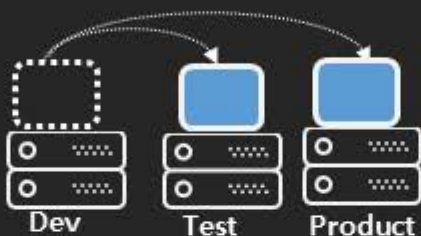
On Demand Delivery

필요한 컴퓨팅 자원을 즉시 제공



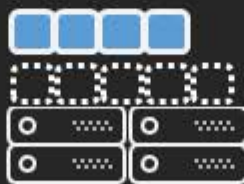
Self Recovery

- 비정상 애플리케이션 재시작
- 노드의 장애 발생시 정상 서버 노드로 자동 재배치



Consistency & Continuous

이미지 기반으로 구성, 배포 효율화
개발과 운영 환경의 일관성



Application Scaling

VM 단위가 아닌 어플리케이션 단위의 오토스케일링



Rolling Update

업그레이드 또는 패치 시
다운 타임은 제로 또는 최소화

Bare Metal



Portable

멀티/하이브리드 클라우드 기반
어플리케이션/서비스 운영



Cloud Native

How does one build apps for the cloud?

- 가상화 기반 IaaS Cloud



Microsoft
Azure

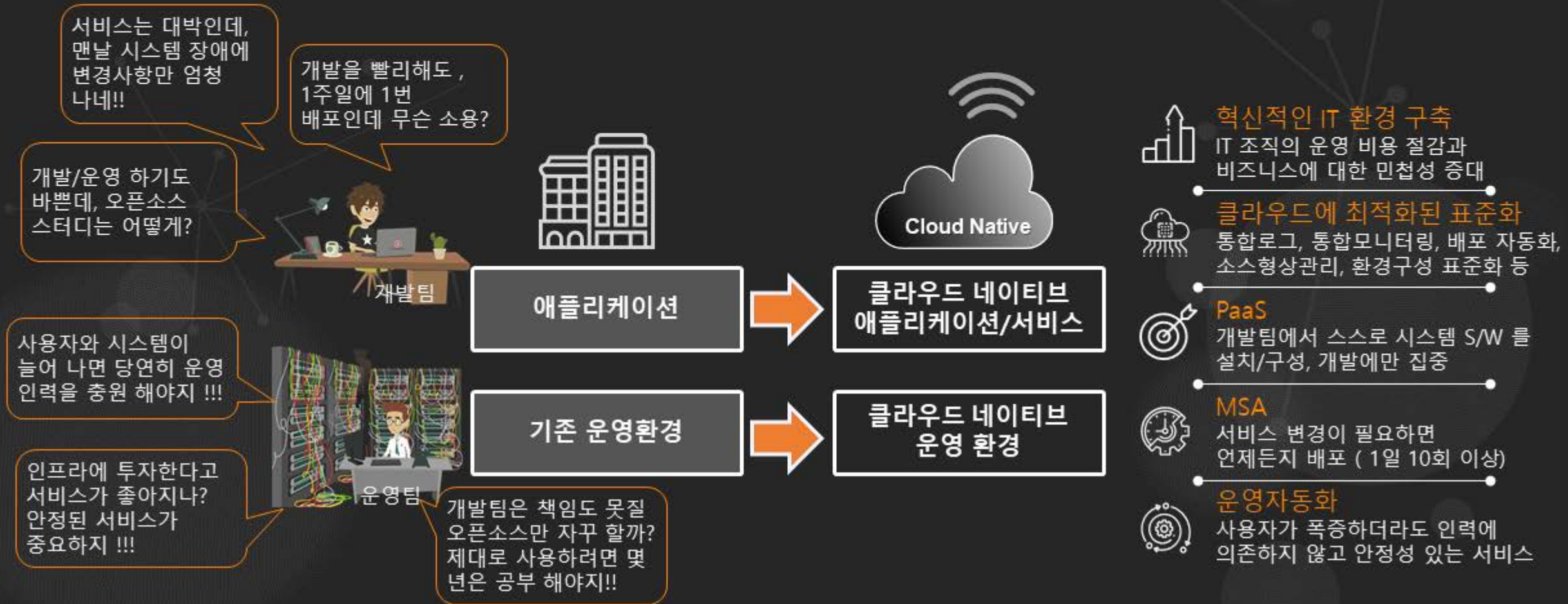


Google Cloud
KVM



클라우드에 대한 개발팀과 운영팀의 고민들

- **Cloud Native Computing**은 클라우드의 특성과 장점을 적용하여 구성된 컴퓨팅 환경으로, 인프라, 플랫폼, 어플리케이션/서비스와 개발, 운영, 관리의 전체 영역을 대상으로 함



공공 분야 - 클라우드 네이티브 정보시스템 구축사업 고려사항

- 클라우드 전환 및 도입 효과를 높이기 위해 단순한 기술 인프라 위주의 클라우드 도입보다
- 클라우드 환경에 최적화된 새로운 형태의 클라우드 네이티브 정보시스템 구축이 필요

클라우드 네이티브 정보시스템 구축을 위한

발주자 안내서



- 사업 추진 방향성과 사업 범위 작성 시 MSA, 컨테이너, 데브옵스 및 CI/CD 구성요소 관련 내용을 포함하여 클라우드 네이티브 사업임을 명시한다.

[그림 5-2] 사업추진방향성작성예시

MSA, 컨테이너 구성요소를 포함하여 사업 추진 방향성 작성

- MSA 기반의 컨테이너 형태로 구현된 공간정보 서비스 가능(공간정보 표준 프레임워크)을 효율적으로 운영 관리하기 위한 개방형 공간정보 플랫폼 구축
- 서비스 수요 증감에 따라 유연하게 컨테이너가 확장 및 축소가 가능한 운영관리 기능 및 컨테이너 동작 여부 모니터링 기능 제공

[출처 : 디지털 권관자원 통합관리시스템 재구축 및 운영 제안요청서, 한국관공공사]

[그림 5-3] 사업범위 작성예시

MSA, 컨테이너, CI/CD 구성요소가 포함된 사업 범위 작성

- 다중화 기반 마이크로 서비스 구축
 - (마이크로 서비스 아키텍처 구축) 컨테이너 관리 기능, API 게이트웨이 관리 기능
 - (전자정부 표준프레임워크 적용) 실행환경 구성, 개발환경 구성, 운영환경 구성, 관리환경 구성
 - (인프라 구축) 인프라 가상화 자동화 구현, HW-SW 구축, 보안관리
- (클라우드서비스) 잘 정의된 API를 통해 콘텐츠를 제공하는 콘텐츠 관리 기능 중심의 시스템 구축
- 마이크로서비스를 독립적으로 개발/배포/관리할 수 있는 프레임워크 제공
 - 컨테이너 관리 : 여러 대의 서버에서 여러 개의 컨테이너를 편리하게 관리하도록 서비스 예시 컨테이너 오케스트레이션 등 자동화 기반의 컨테이너 배포 구현
- 마이크로서비스 구현을 위한 가상화/자동화 환경을 제공
 - 가상화 : 물리적/논리적 서버 클러스터 구성을 통해 시스템 가용성 향상 및 가상 서버 복제 및 수정 확장을 통해 시스템 확장성 확보
 - 자동화 : 서비스 요청관리, 수요관리, 변경관리 등 사용자의 서비스 요청에 대한 해당 서비스를 사용자에게 제공하기 위해 다음과 같은 자동화된 서비스 제공관리 환경 구축

[출처 : 디지털 권관자원 통합관리시스템 재구축 및 운영 제안요청서, 한국관공공사]

[그림 5-5] 상세요구사항작성예시

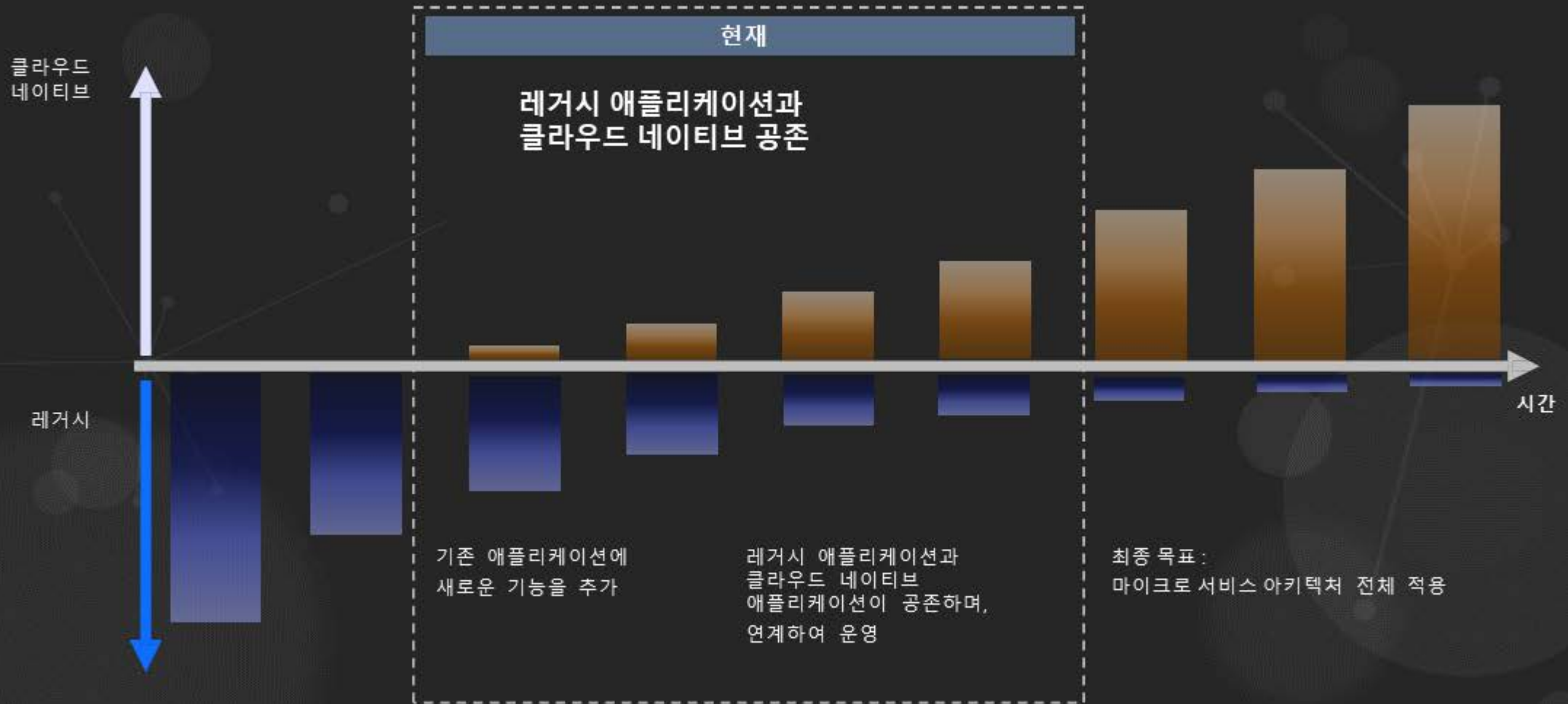
요구사항 분류	클라우드 서비스 요구사항
요구사항 고유번호	CSR-003
요구사항 명칭	컨테이너 기반 서비스 예시 및 오케스트레이션 구현
요구사항 상세 설명	<ul style="list-style-type: none"> 여러 대의 서버에서 여러 개의 컨테이너를 편리하게 관리하도록 서비스 예시 가능 제공 컨테이너를 적절한 서버에 배포하고 상태를 유지하기 위한 스케줄링 여러 대의 서버를 1대의 서버처럼 관리하고, 가상 네트워크를 이용해 접근하기 위한 클러스터링 컨테이너의 IP/포트정보를 서비스 레지스트리에 저장하며, 동적으로 변화하는 리소스의 위치를 API 게이트웨이가 검색하기 위한 서비스 디스커버리 기능 제공 API 요청에 대한 최적의 경로를 지원하기 위한 다양한 API 라우팅 구현 서비스 간 부하 분산을 위한 로드밸런싱 오로케스트레이션 시 서버 수 지정, 서버의 사양 정의, 서버 실행 시까지의 워밍업 시간 지정 등 트래픽 집중에 서버 스토리지, 네트워크 등 인프라 자원의 자동 확장 및 축소를 자동화하여 서비스 상태에 따른 적정 서버 유지를 통해 유연한 서비스를 제공하도록 오로케스트레이션 지원 특정 서비스에 오류가 발생하거나 실행 실패 시 신속하게 이전 버전으로 되돌아가도록 복구(Rollback) 기능 지원 표준화된 로그 이벤트 수집 및 분석, 서비스 간 호출 추적 및 성능 관리 등 로깅 및 로그 분석 등

[출처 : 나라장터, 클라우드 네이티브 관련 제안요청서 참조]

요구사항 분류	클라우드 서비스 요구사항
요구사항 고유번호	CSR-004
요구사항 명칭	API 게이트웨이 관리 기능
요구사항 상세 설명	<ul style="list-style-type: none"> API 호출을 위한 토큰 발급 및 인증, 엔드포인트별 API 호출 인증 및 인가, 접근 정책제, 특정 클라이언트와 API 호출 불허에 의한 접근제한 기능 등 API 인증 및 인가 처리 동일 API를 클라이언트나 마이크로서비스에 따라 다른 엔드포인트를 통해 서비스를 제공하도록 API 라우팅을 지원하고, 동일 API를 여러 개의 클라이언트/마이크로서비스 별로 엔드포인트 제공 로그, 인증 등 공통 기능을 중복 개발 또는 처리하지 않도록 요청과 응답의 표준화 및 공통 로직 처리 동일 API를 HTTP, REST, XML, 웹 서비스 등 클라이언트와 마이크로서비스별로 상이한 프로토콜로 서비스하기 위한 프로토콜 변환 처리 동기, 비동기 등 API를 호출하는 메시지 패턴을 변경할 수 있도록 메시지 호출 패턴 변환 기능 제공 호출 확률, 전송 용량, 네트워크 대역폭 등 서비스 레벨을 클라이언트나 마이크로 서비스별로 조정할 수 있도록 QoS(Quality of Service) 설정 기능 제공 API 호출 패턴 분석, API 호출 실행 및 접근 상태 분석, 요청 IP/클라이언트/일시 등 API 호출에 대한 로깅 및 모니터링 등

[출처 : 나라장터, 클라우드 네이티브 관련 제안요청서 참조]

레거시와 클라우드 네이티브 애플리케이션 전환



Source : IBM



openmaru