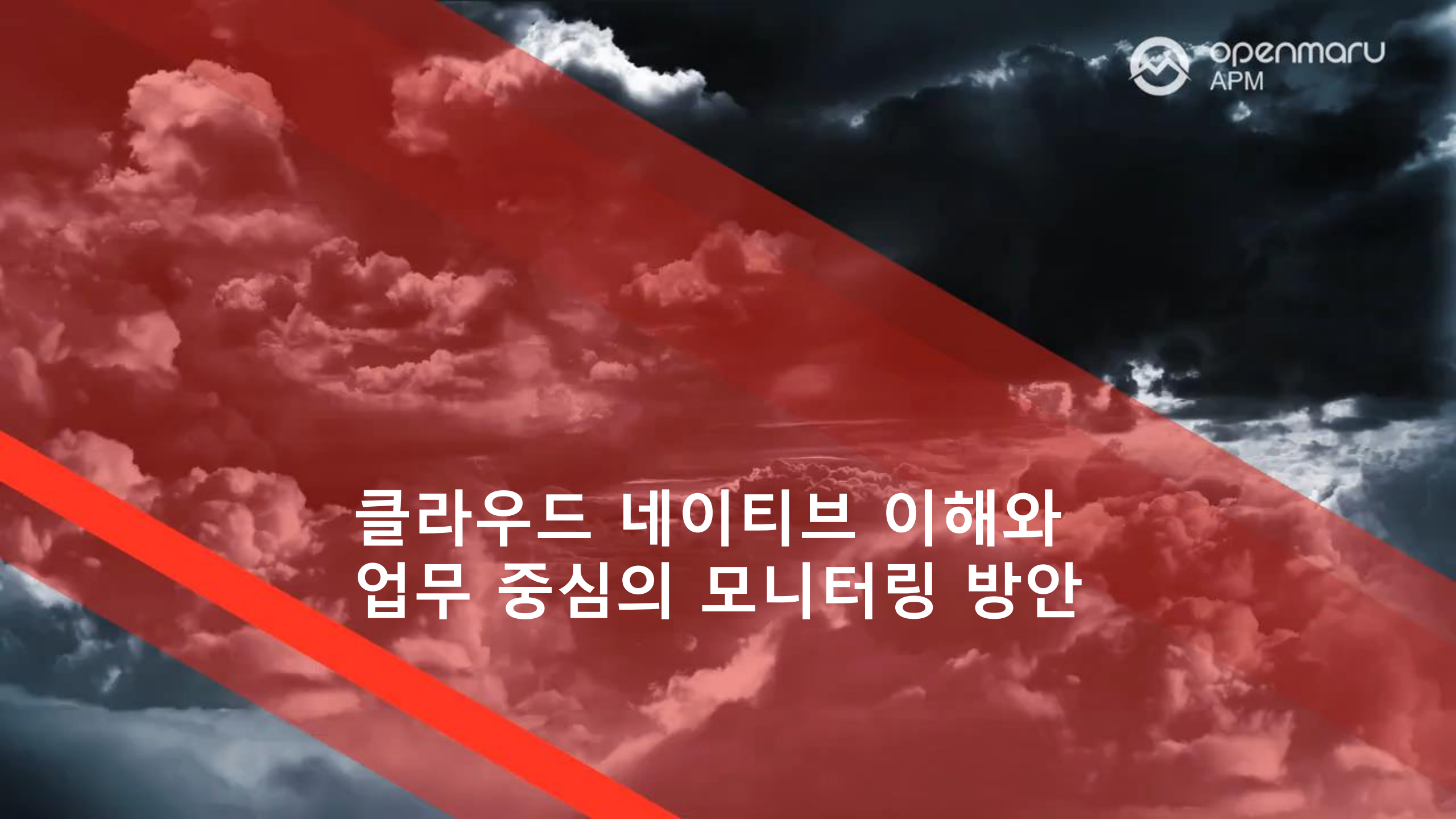

openmaru
APM
클라우드 네이티브 이해와
업무 중심의 모니터링 방안

디지털플랫폼정부는 '클라우드 네이티브'로 구축하는 게 핵심이다

전자신문 etnews Conference allshowTV Engl

경제·금융 전자·모빌리티 통신·미디어·게임 소재·부품 SW·보안 산업·에너지·환경

[기획]행안부, 클라우드 네이티브 확산 위한 지원사업 추진

발행일 : 2022-06-29 16:00 지면 : 2022-06-30 18면

<클라우드 성숙도 단계>

클라우드 네이티브는 클라우드 성숙도 단계 중 최고 단계로, 클라우드의 기능과 장점을 최대한 활용해 애플리케이션을 구축·실행하는 것을 의미한다.

공공클라우드 전환 로드맵 손질...부처별 추진·2030년 완료

류태웅 | 입력 2023. 5. 3. 15:32 | 수정 2023. 5. 4. 09:01

0

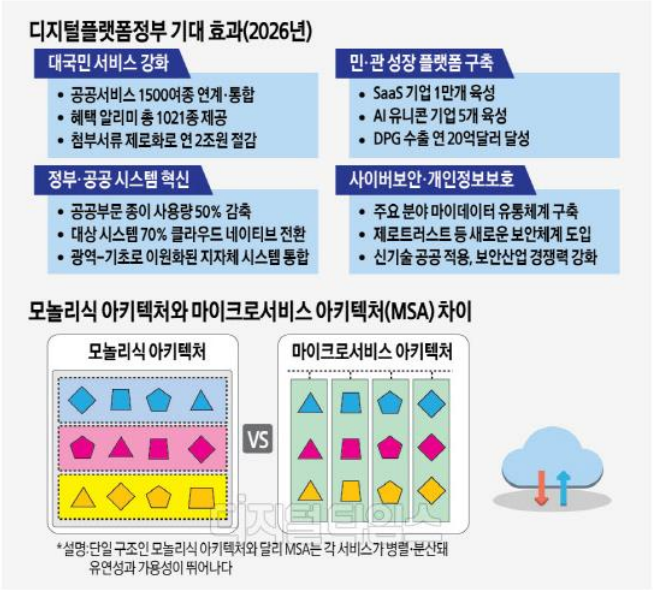
공공 클라우드 네이티브 전환 로드맵

추진 배경	•정부 재정 투자 방향 변화 •보안인증제 개편 •신기술 보편화 등
추진 환경	•행정안전부 전환 사업 예산 축소 •정부 정책 '민간 클라우드 우선 이용' '클라우드 네이티브 우선 적용'으로 발전 등
추진 방향	•범정부 정보자원 등록·관리시스템에 등록된 모든 시스템의 클라우드 네이티브 전환
전환 기간	•2024년부터 2030년까지 7개년 추진

[테크&포커스] 정부·공공 시스템도 `클라우드 최우선 도입 전환`... `디플정`의 도전

입력: 2023-04-16 16:07 팽동현 기자

가정부 핵심과제 청사진 공개
민첩한 개발·유연한 확장 가능
초거대 AI 등 디지털기술 도입
홈택스 등 사이트 통합도 추진



디지털플랫폼정부는 초거대AI를 비롯한 디지털 기술을 적극 도입, 칸막이를 없애고 '원팀 정부'로 거듭나는 것을 목표한다. 이를 위한 공공 플랫폼을 단순 클라우드 전환이 아닌, **클라우드를 클라우드 답게 쓸 수 있도록 '클라우드 네이티브'로 구축하는 게 이번 정책의 핵심**이다. 공공SW(소프트웨어)사업 고질병을 극복할 해법을 제시하면서 각종 미래 IT산업 육성의 요람이 될 것으로 기대된다.

조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

openmaru

조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

openmaru

openmaru

"메르세데스 벤츠는 이제 자동차 기업이 아닌 소프트웨어 기업이며 자동차는 궁극의 웨어러블이다."
- 올라 켈레니우스, 메르세데스 벤츠 그룹 대표

"우리는 클라우드 네이티브로 기술을 재구축하면서 조직 운영 방식도 바꿨다."
- 넷플릭스 클라우드 부문 부사장 이즈라일레브스키

"골드만삭스는 정보기술(IT) 회사다."
- 골그만삭스 로이드 블랭크파인 (CEO)

"인간 중심의 디지털 전략이 미래 리테일 산업의 성공의 열쇠가 될 것입니다."
- Kevin Johnson, CEO of Starbucks

openmaru

Development Process

WATERFALL

AGILE

DEVOPS

Application Architecture

MONOLITHIC

N-TIER

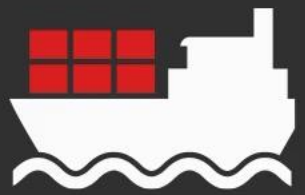
MICROSERVICES

Deployment & Packaging

PHYSICAL SERVERS

VIRTUAL SERVERS

CONTAINERS



Application Infrastructure

DATA CENTER

HOSTED

CLOUD

SoR 에서 SoE 로 전환

	Systems of Record (SoR)			Systems of Engagement(SoE)		
	Mainframe	Mini	PC	Internet	Mobile/Cloud	Connected
Time Frame	1960 - 1975	1975-1992	1992-2001	2001-2009	2010-2015	2016-
Data Types	Batch	Dept Process	Documents	Web Pages	User Interactions	IOT / AI
First Movers	IBM, Unisys	DEC, Compaq	Microsoft, Dell, IBM	Google, Microsoft	Facebook, Amazon, Apple	Airbnb, Uber

데이터를 기록하는 시스템	*역할*	연결·관계·상호작용을 만드는 시스템
신뢰성/안전성/정확성/데이터 중심	*포커스*	즉시성/유연성/사용자 중심
Waterfall	*어프로치*	DevOps & Agile
Java & . NET	*프레임웍*	경량/다양한 프로그램 언어
독점 기술 중심/ 물리&가상 환경	*기술*	오픈 / 클라우드
모노리식	*아키텍쳐*	마이크로 서비스
웹기반	*어플리케이션*	Mobile
일괄/실시간 처리	*트랜잭션*	리얼타임, 탄력적인 인프라

머신 중심에서 애플리케이션 중심 인프라로 변화

- 컨테이너화는 데이터 센터를 머신 중심에서 애플리케이션 중심으로 전환
  - 개발자와 운영팀에게 서버와 운영 환경에 대한 세부 사항을 추상화
  - 운영 중인 애플리케이션과 개발자에 미치는 영향을 최소화하면서 새로운 하드웨어 또는 운영 환경을 업그레이드하여 인프라팀에게 유연성을 제공
  - 서버의 CPU와 메모리 정보 뿐만 아니라 애플리케이션과 관련한 매트릭을 연결하여 오토 스케일링

Machine Centric Infrastructure

Application Centric Infrastructure

조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

openmaru

조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루** 를 검색하세요.

openmaru

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

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)

Native

- 네이티브(Native)의 사전적 의미는 ‘선천적인’, ‘본래’ 등이다.
- 클라우드 네이티브는 "클라우드가 ‘클라우드 다울 수 있도록’ 애플리케이션을 구축, 실행하는 방식”

'어린이 또는 성인이 되어 언어를 배운 것이 아닌 태어나서 부터 특정 언어를 사용해 온 사람'

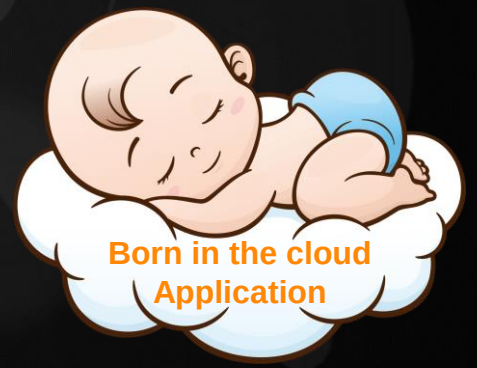
네이티브 스피커

'어린이 또는 성인이 되어 스마트폰을 접한 것이 아닌 유아기부터 스마트폰을 사용해 온 사람'

스마트폰 네이티브

‘애플리케이션을 계획/설계할 때부터 클라우드 특징과 장점을 기반으로 개발/운영'

클라우드 네이티브



컨테이너는 클라우드 에서 Java 와 같이 벤더 종속성 해제

openmaru

2000 년 – Java 를 통한 Vendor Lock-In 해제

OS	HP-UX	AIX	Solaris	Linux	Windows
CPU	Intanium	Power	Sparc	X86	X86
Vendor	HP	IBM	Oracle	The Linux Foundation	Microsoft

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

Hypervisor	Xen Project	KVM	Hyper-V	Xen Project	Xen Project
Public Cloud	aws	Google Cloud	Microsoft Azure	Oracle	IBM Cloud

How does one build apps for the cloud?

openmaru

KVM

Google Cloud

Xen Project

aws

Hyper-V

Microsoft Azure

Xen Project

Alibaba Cloud

Xen Project

IBM Cloud

Hypervisor

Public Cloud

Virtual Machine

Write once, run anywhere?

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.

Source:https://github.com/cncf/toc/blob/master/DEFINITION.md#%ED%95%9C%EA%B5%AD%EC%96%B4

Cloud Native Computing Foundation

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

- Platinum members:

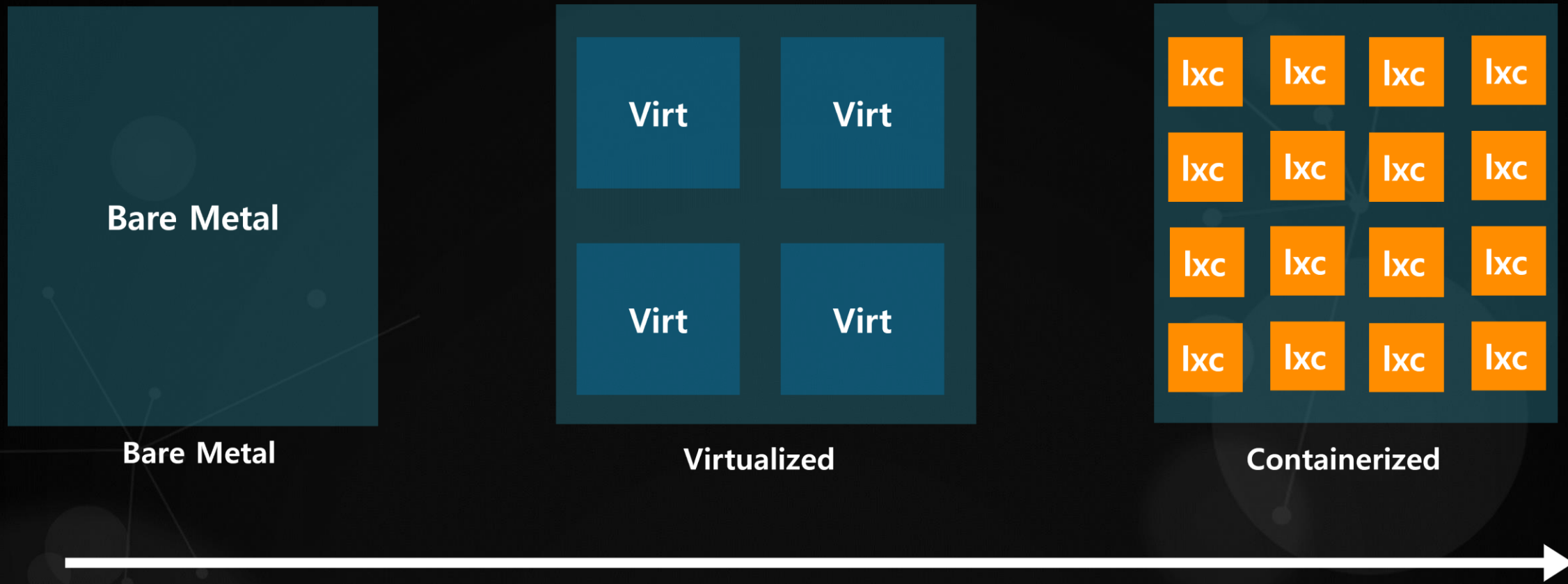
조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

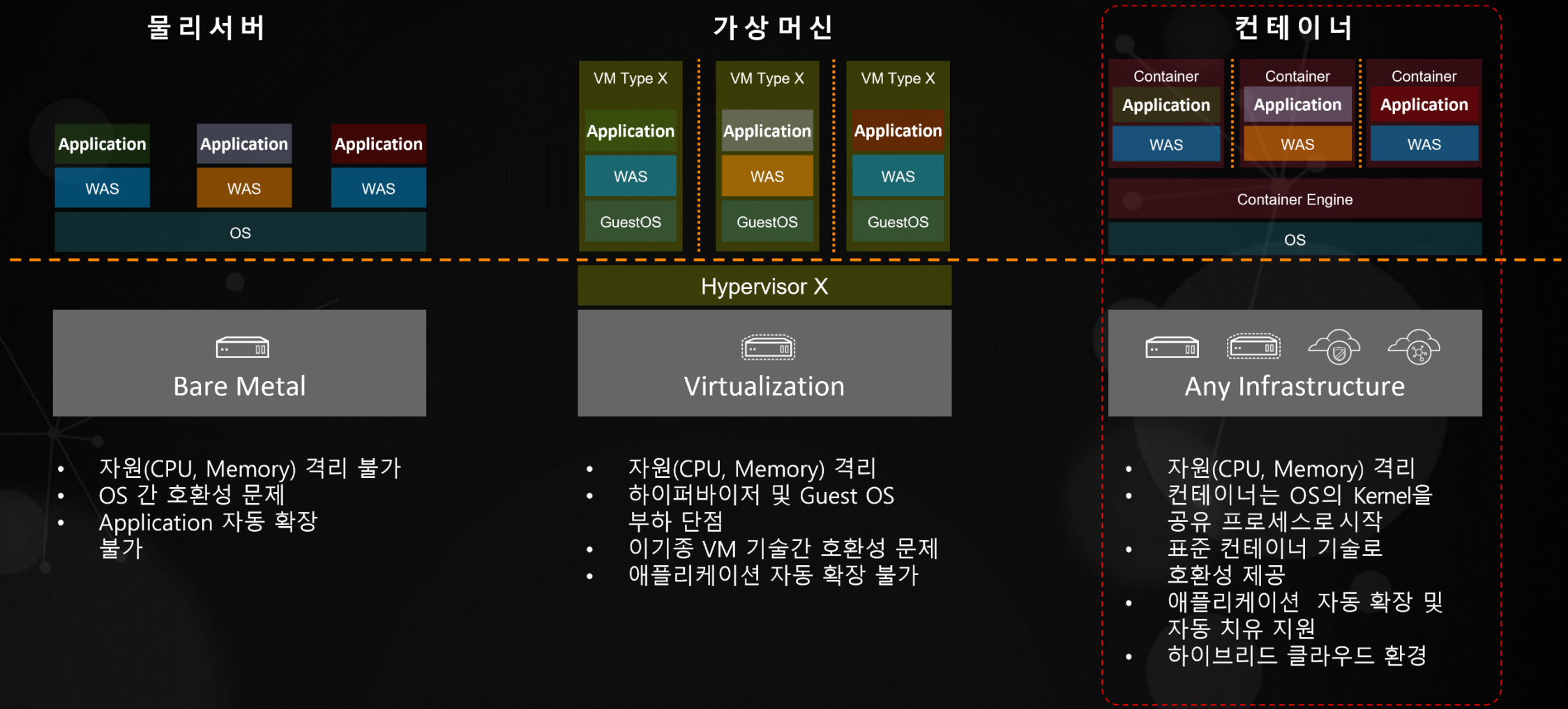
openmaru

Evolution of Infrastructure Architectures



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

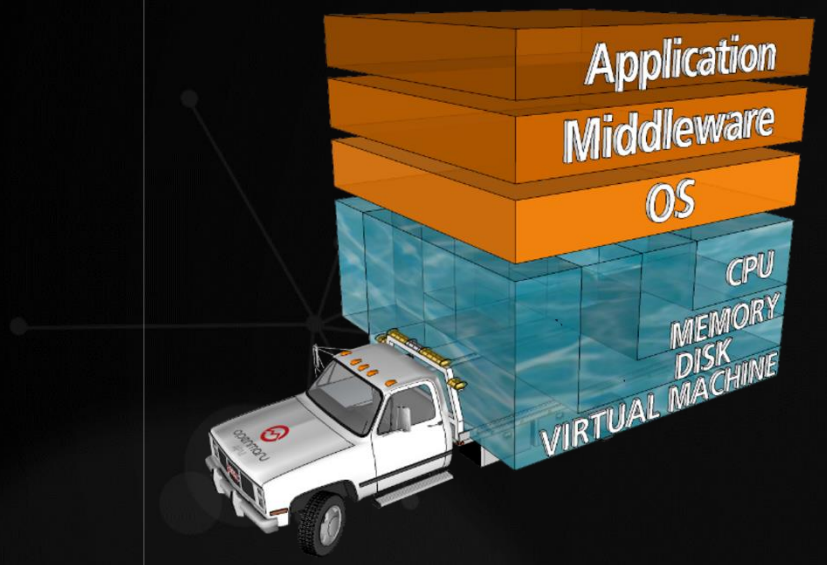
시작 시간 - Containers vs. VMs

- 하드웨어 가상화는 CPU, 메모리, 하드 디스크 등의 하드웨어를 가상화하고 있기 때문에 하드웨어와 OS 부팅이 필요 (부팅에 분 단위 시간 소요)
- 컨테이너 형 가상화에서는 컨테이너 부팅 시 OS가 이미 시작된 상태 (프로세스 시작에 필요한 초 단위 시간 소요)

가상서버의 고질적인 문제점

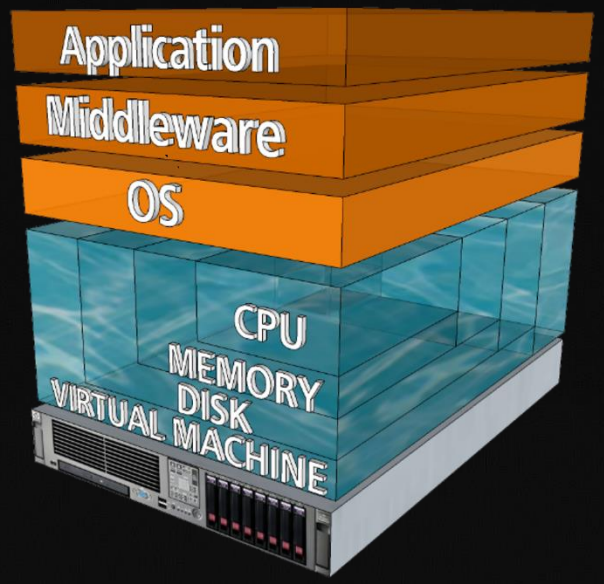
거대한 이미지 사이즈

- VM 을 템플릿 관리는 하지만 사이즈가 커서 재사용성을 높이기 어려움



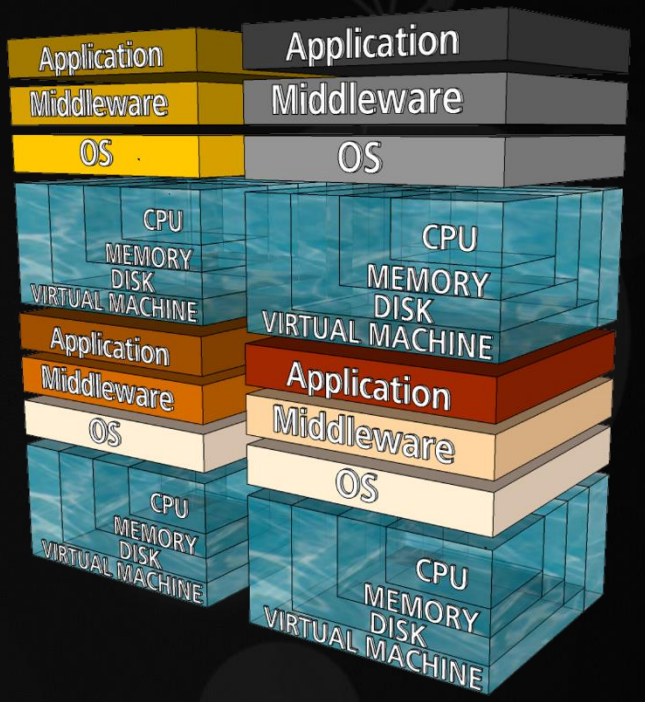
느린 시작 시간

- 부팅 시 Hypervisor – OS- 미들웨어 – 애플리케이션까지 실행 되어야함



VM 간의 환경 불일치

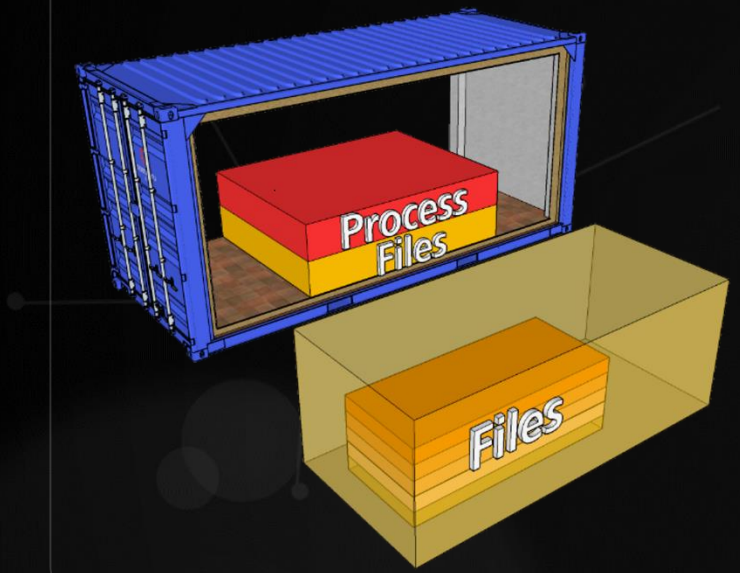
- VM 생성 후 개별로 변경 사항을 관리하기 때문에 VM 간 구성이나 환경이 불일치



컨테이너를 통한 가상서버 문제점 해결

작은 이미지 사이즈

- 컨테이너는 레이어 개념으로 이미지에 파일을 추가/삭제하여 관리함
- 레이어 사이즈를 최적화하여 이미지 사이즈를 최소화



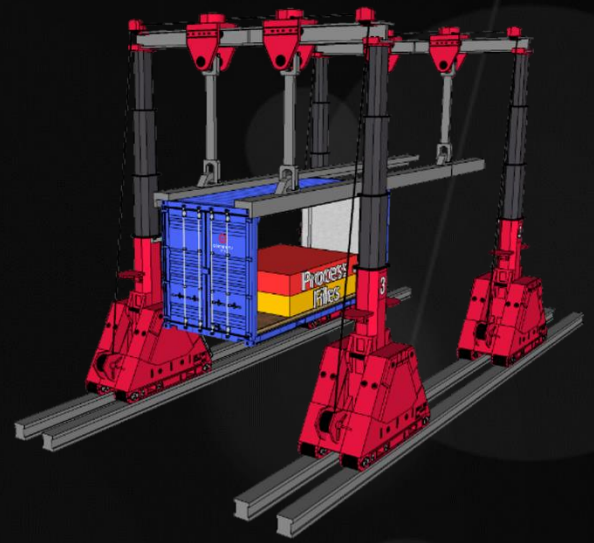
빠른 시작 시간

- 컨테이너는 분리된 프로세스 형식으로 OS 부팅이 필요 없기 때문에 부팅 시간을 최소화 할 수 있음

Container = Process

높은 이동성(Portability)

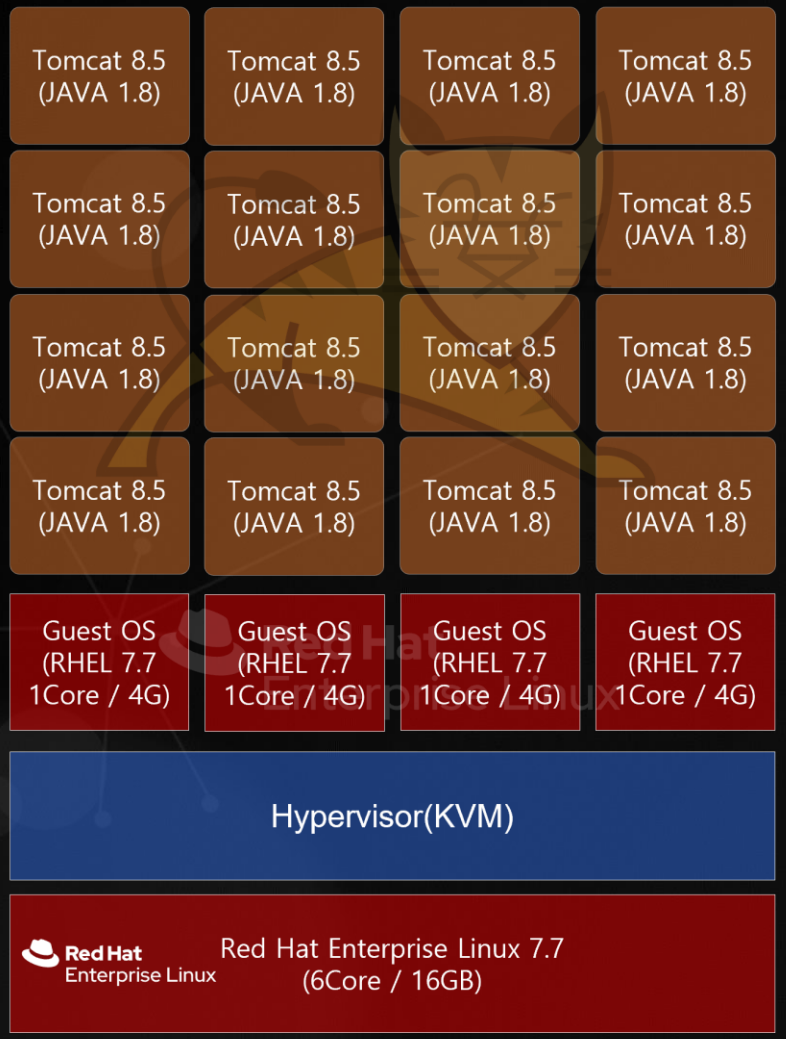
- 애플리케이션에 필요한 라이브러리나 의존 파일들을 이미지에 포함하기 때문에 환경에 의한 발행되는 문제가 거의 없음



가상화와 컨테이너 집적도 비교

openmaru

가상화 - 16개 Tomcat 인스턴스



컨테이너 - 40개 Tomcat 인스턴스

조달청 디지털서비스몰 등록

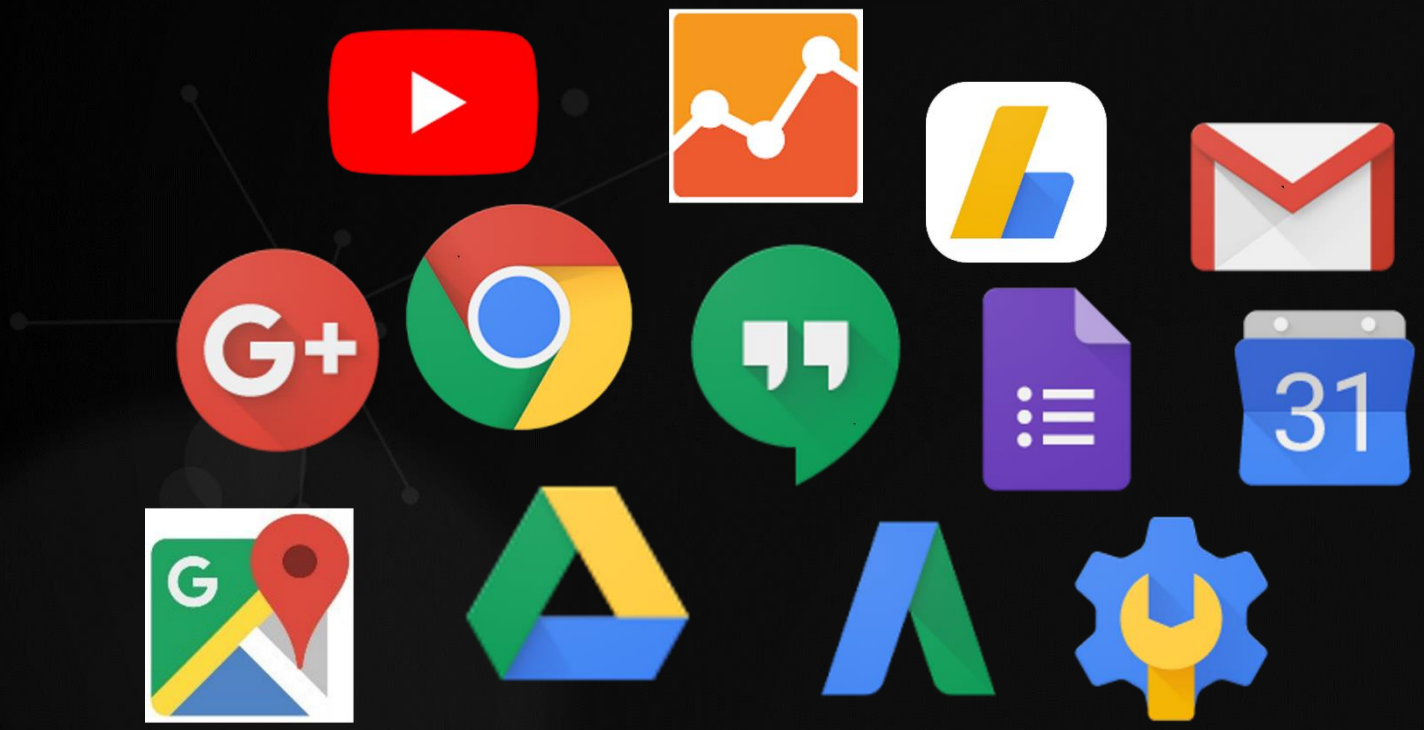
오픈마루

디지털서비스몰에서 **오픈마루** 를 검색하세요.

openmaru

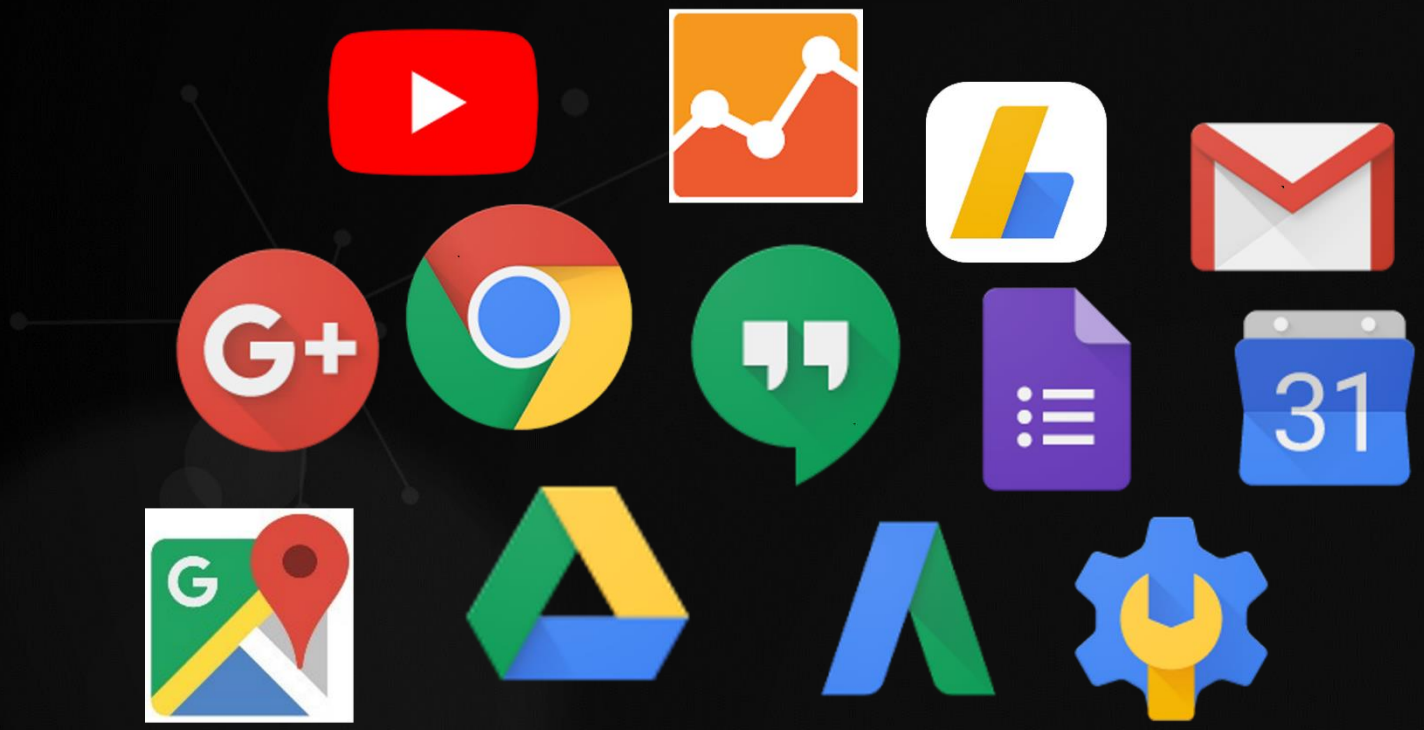
Google 의 모든 서비스는 [ ] 에서 실행

- Gmail , 검색, 지도 ...
- MapReduce , GFS , Colossus ...
- Google Compute Engine 가상 머신도 [ ] 에서 실행!
- 매주 20 억개 이상의 [ ] 를 실행 중



Google 의 모든 서비스는 컨테이너 에서 실행

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



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 생성**합니다. **지난 10여 년간 프로덕션 환경에서 컨테이너화된 워크로드를 실행하는 방법에 관해 많은 경험**을 쌓으면서 Google은 커뮤니티에 계속 이 지식을 공유해 왔습니다.

초창기에 cgroup 기능을 Linux 커널에 제공한 것부터 **내부 도구의 설계 소스를 Kubernetes 프로젝트로 공개**한 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.

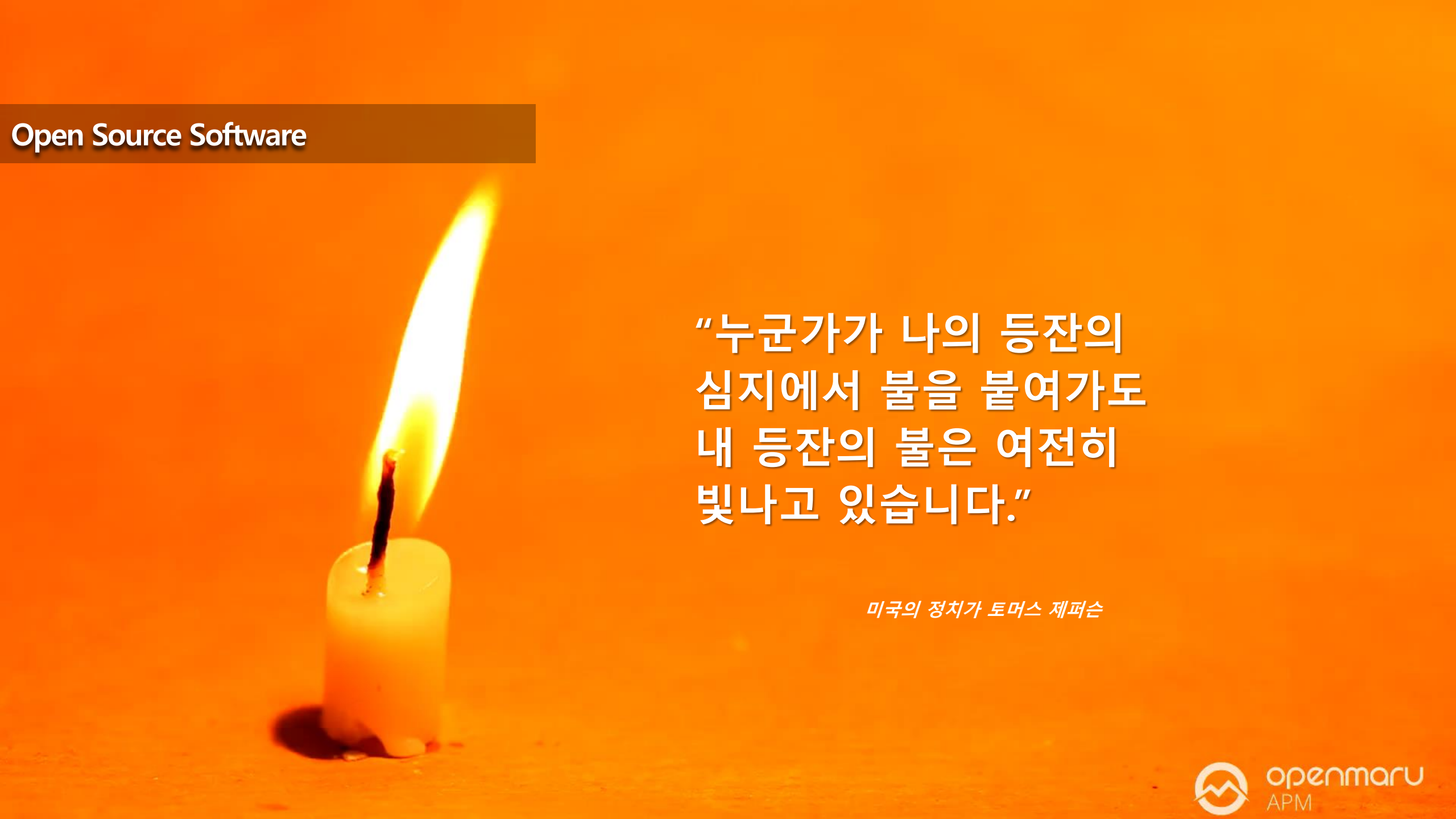
Source - https://cloud.google.com/containers/

About Kubernetes

- 쿠버네티스(K8s)는 컨테이너화된 애플리케이션을 자동으로 배포, 스케일링 및 관리해주는 오픈소스 소프트웨어
- 쿠버네티스”, “쿠베르네테스”,“K8s”, “쿠베”, “쿠버”, “큐브”라고 부르며
- Go로 작성된 오픈 소스 , 오픈소스 S/W (Apache License 2.0) 라이선스
- 리눅스 재단 (Linux Foundation)산하 Cloud Native Computing Foundation (CNCF) 에서 관리
- 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*

Open Source Software



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

미국의 정치가 토머스 제퍼슨

openmaru
APM

조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

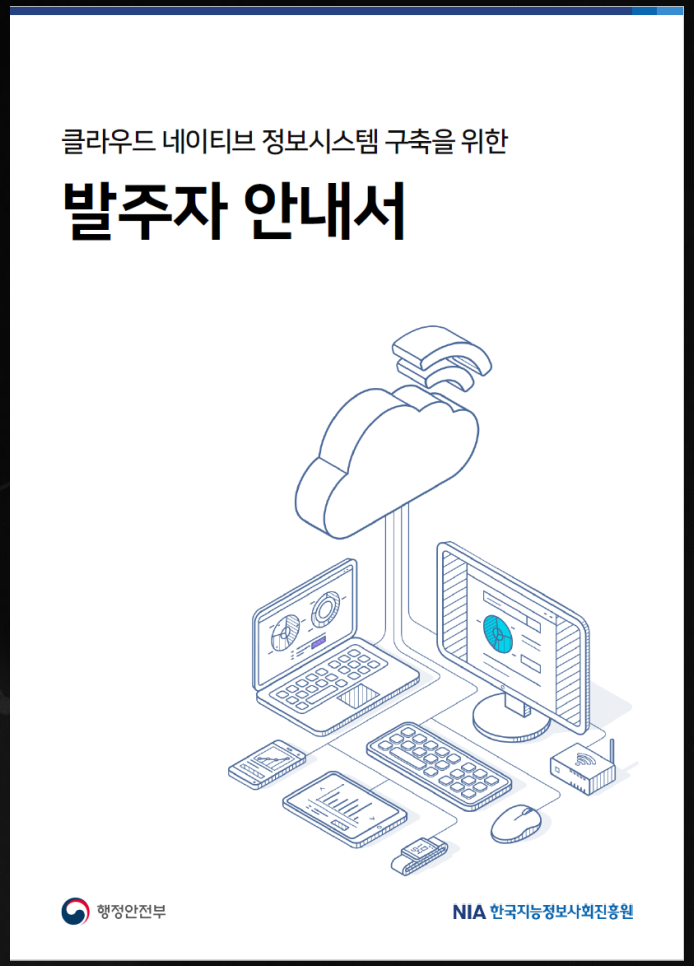
openmaru

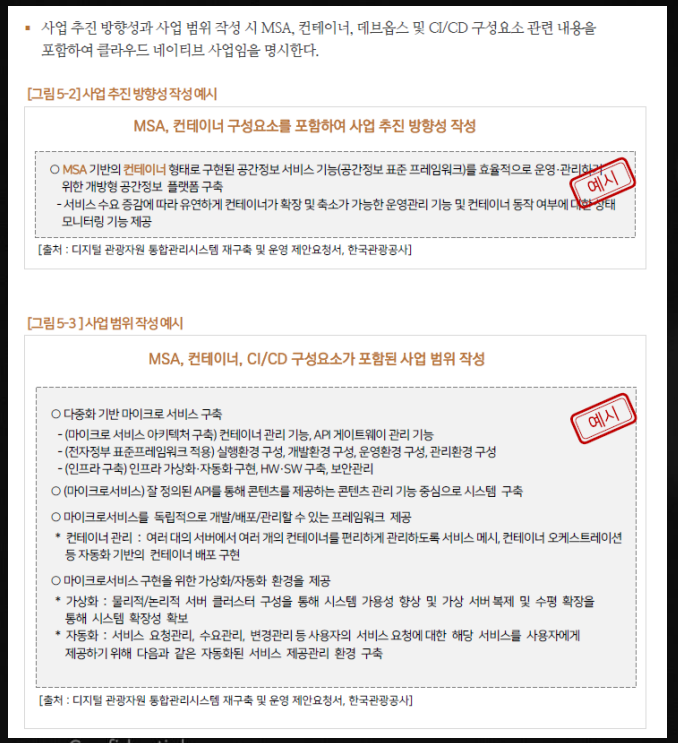
Cloud Native 시대 2016 년 ~

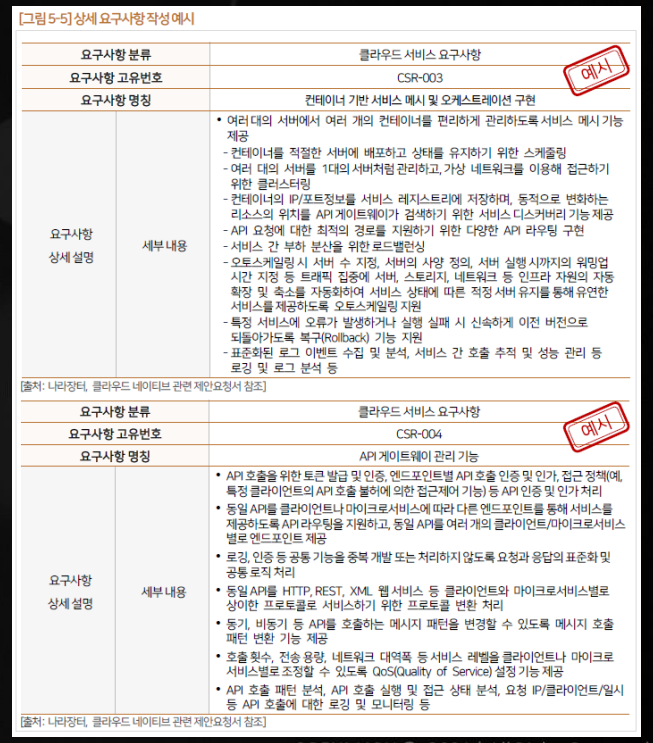
- 2016 년 1 월에 정식 출범 한 Cloud Native Computing Foundation (이하 CNCF)는 진짜 클라우드 기술을 오픈소스로 해결하는 하는 것을 목표
- 애플리케이션을 실행하는 데 필요한 최적의 인프라 제공
  - 개발 한 것을 "즉시" "안정적으로" 제공 (비즈니스 우위)

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 정의

- 클라우드 네이티브(Cloud Native)는 “클라우드의 장점을 최대한 활용하여 정보시스템을 구축 및 실행하는 환경”
- 2015년 최초로 클라우드 네이티브라는 용어를 사용한 리눅스 재단은 CNCF(Cloud Native Computing Foundation) 재단을 설립하여 클라우드 네이티브 관련 기술을 정의하고 오픈소스를 관리

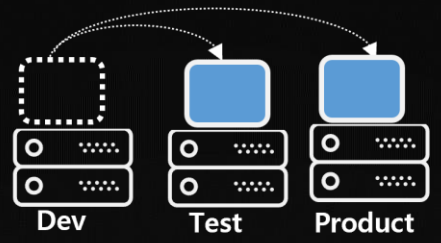
Source - 클라우드 네이티브 정보시스템 구축을 위한발주자 안내서

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

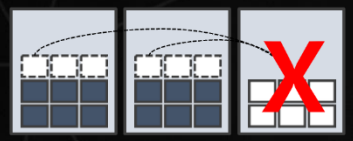


Consistency & Continuous

- 이미지 기반으로 구성, 배포 효율화
  개발과 운영 환경의 일관성

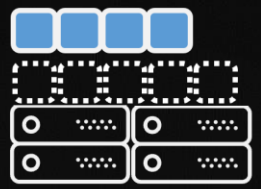
Rolling Update

- 업그레이드 또는 패치 시
  다운 타임은 제로 또는 최소화



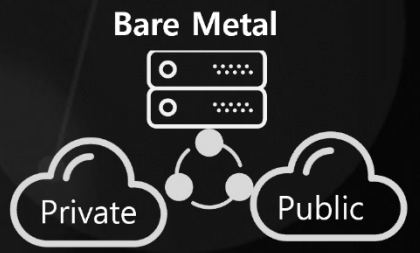
Self Recovery

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



Application Scaling

- VM 단위가 아닌 어플리케이션
  단위의 오토스케일링



Portable

- 멀티/하이브리드 클라우드 기반
  어플리케이션/서비스 운영

조달청 디지털서비스몰 등록

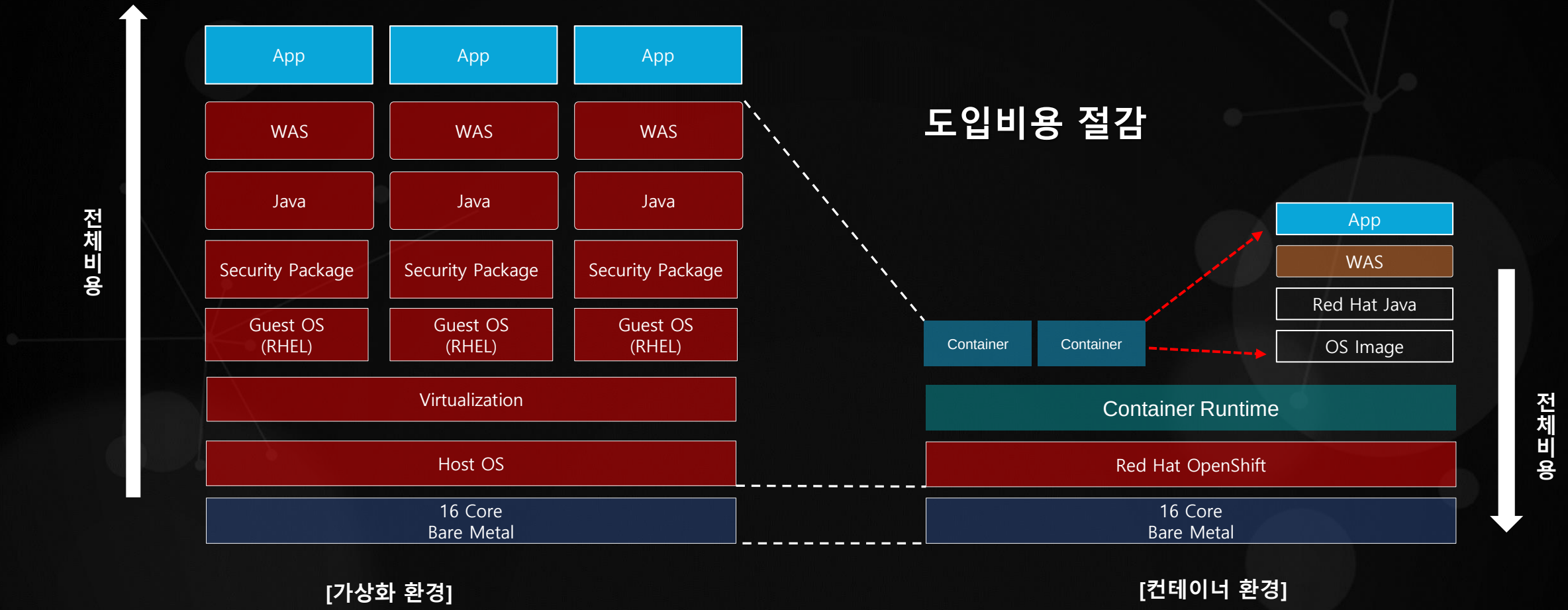
오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

openmaru

가상화 VS 컨테이너 비교 – 비용적인 측면

- 가상화 대비 Guest OS 유지보수, 라이선스, 관리비용 제거
- 서버 접근제어를 비롯한 보안 솔루션 제거



아직도 WAS BMT 나 POC를 하시나요?

- WAS 의 가용성/확장성/성능/신뢰성 등의 기능은 플랫폼으로 이전
- 더 가볍고 더 빠르고, 자동화에 친화적인 WAS 로 전환

WAS 제품 BMT

ORACLE WEBLOGIC

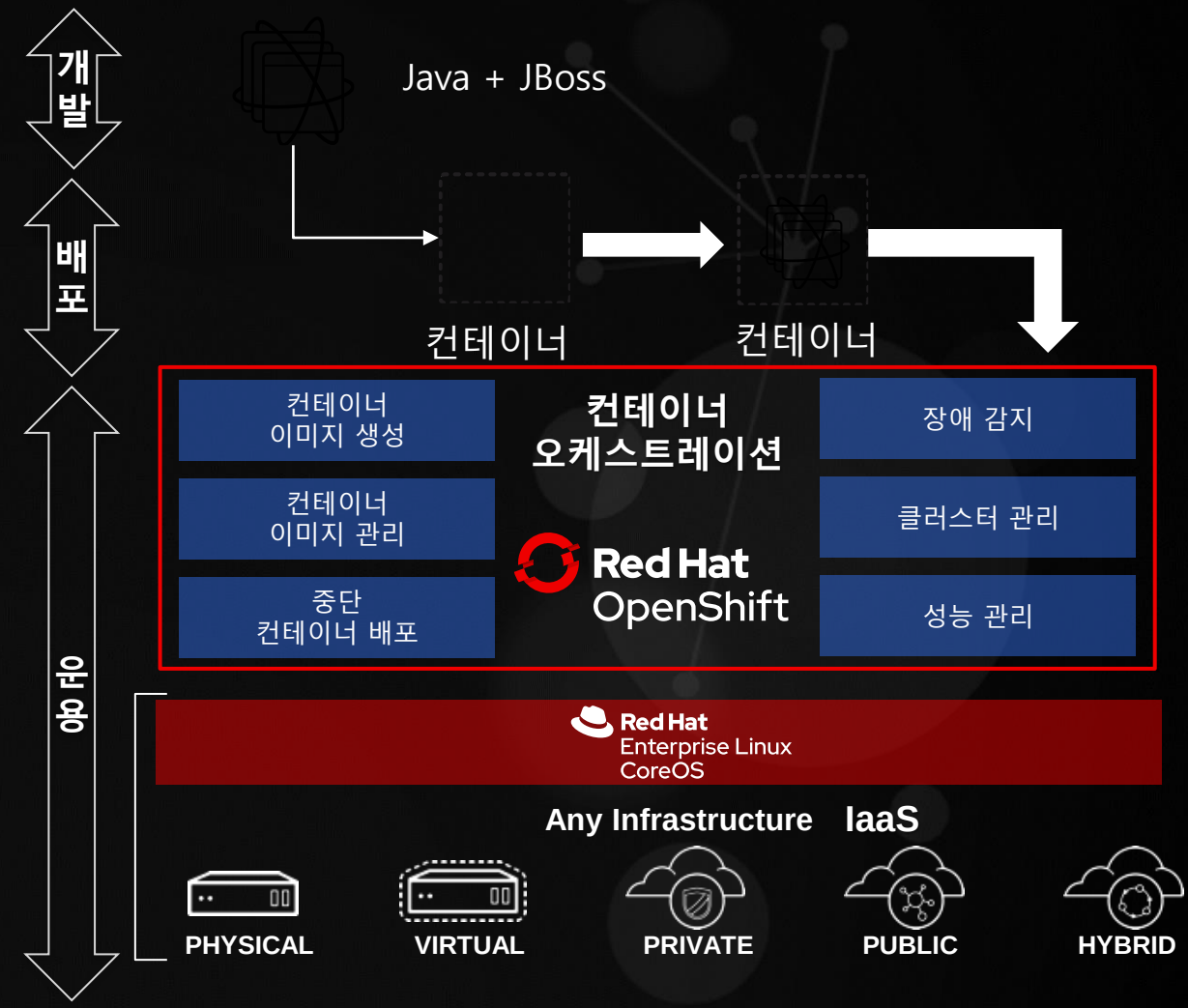
JBoss by Red Hat

WebSphere

Tmax Web Application Server JEUS

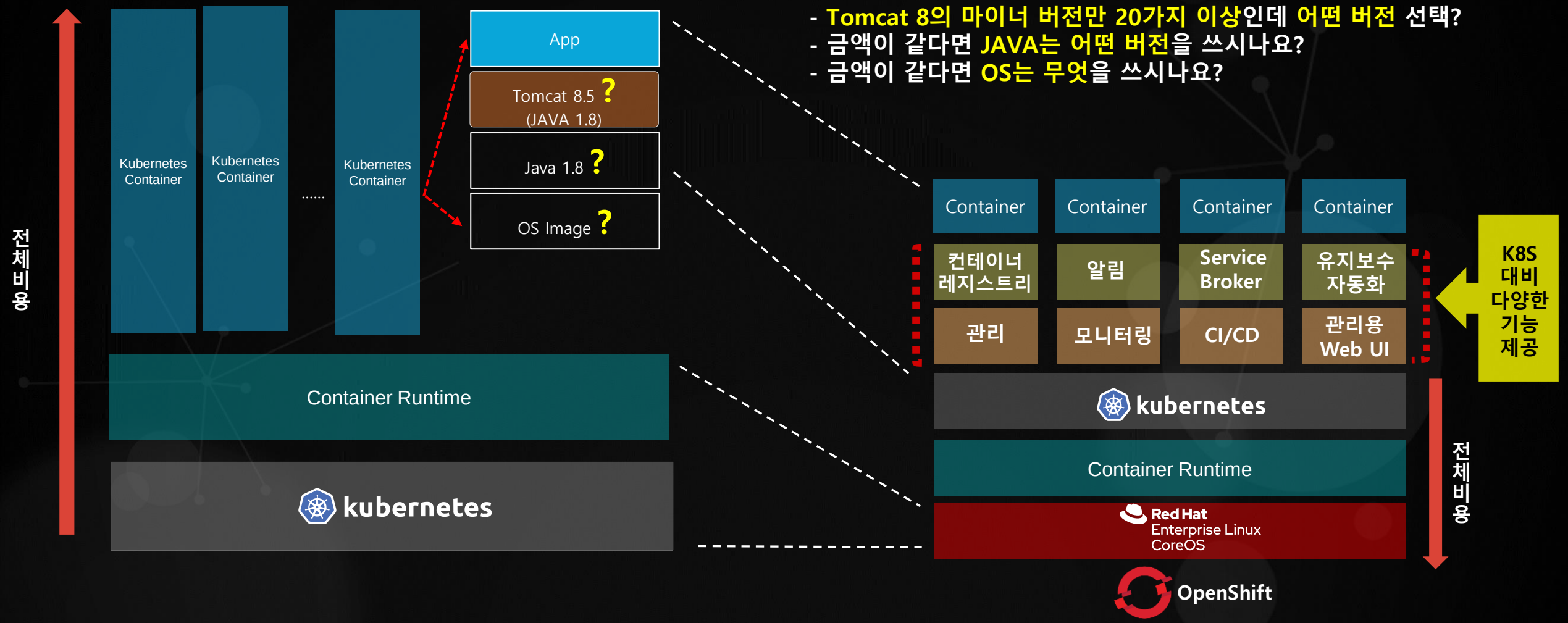
RASP 에 의한 WAS평가

- 신뢰성 (Reliability):
  - 과부하 테스트
- 가용성 (Availability):
  - 일부 장애 발생시 전체 시스템 영향 최소화
- 확장성 (Scalability):
  - 자원 추가에 따른 선형적인 성능 개선
- 성능 (Performance):
  - TPS, 응답시간 등 평가
- 보안 기능 (Security)
  - (RBAC 등)
- WAS 도구 평가 (Manageability)
  - Admin Console



운영 환경에 부족한 Kubernetes vs 완벽한 OpenShift

Openshift는 Tomcat/JAVA/OS 이외에도 많은 오픈소스 유지보수 제품들이 포함되어 있습니다.



조달청 디지털서비스몰 등록

오픈마루

디지털서비스몰에서**오픈마루**를 검색하세요.

openmaru

클라우드 전환에 따른 APM 요구사항의 변화

클라우드에서는 APM 모니터링 대상이 WAS에서 컨테이너로 변경

- 컨테이너 단위로 WAS에 대한 확장/축소, 장애 복구 , 업그레이드, 패치 작업하여 WAS와 함께 컨테이너를 모니터링해야 합니다.
- 기존 물리서버나 가상서버와는 달리 컨테이너는 휘발성으로 상태를 가지고 있지 않습니다.

WAS용 APM

운영 요구 사항	사람의 수작업에 의한 운영		
WAS증설과 축소	운영자	수작업	직접확인
WAS 장애	운영자	수작업	직접확인
애플케이션 배포	운영자	수작업	직접확인
WAS 운영관리	운영자	수작업	직접확인
WAS 업그레이드	운영자	수작업	직접확인

새롭게 추가된 APM 요건

- WAS는 물론 컨테이너 모니터링
- 자동장애복구/자동확장축소 모니터링 정보 제공
- 폐기된 컨테이너 정보 제공
- OpenShift 관리 정보와의 통합

컨테이너용 APM

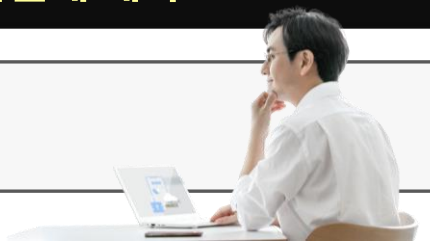
클라우드에서 규칙의 의한 운영 자동화		
클라우드	부하에 따른	자동 확장/축소
클라우드	자동감지	자동 장애복구
클라우드	무정지 서비스	자동 배포
클라우드	컨테이너로	WAS 관리
클라우드	이미지에 의한	WAS 업그레이드

머신러닝 기반 장애 경고 알람

OPENMARU APM은 머신러닝 기반으로 임계치에 도달하기 전에 이상 징후를 사전에 예측

고려사항

- ⊘ 사용자가 다양한 성능과 장애 정보들의 임계치나 추이 파악이 어려움
- ⊘ APM에서 제공하는 모든 정보를 확인할 수 선재적 대응과 장애와 성능관리가 어려움



방안

- 실시간 머신러닝(통계분석)을 통해 앞으로 몇 분 후에 관리자가 설정한 임계 값에 도달할 것이라는 예측 이벤트를 통보합니다.
- APM은 현재 상태 모니터링 및 추세분석을 통해 임계값 도달 시점을 예측하여 이벤트를 통보 한다.

Java VM의 Heap 사용량 증가 추이를 분석하여 GC 임계값 초과 시점에 대한 정보 제공

장애경고 알람 대상 항목

항목	이벤트 발생 항목
WAS	• JVM Heap Usage • APDEX • Error Rate • JVM Perm Usage • Database Response Time • Pending Transactions • GC Usage • Database Connection Pool Usage
WEB	• WEB Traffic • Worker Usage
System	• Memory Usage • Disk Usage • Memory Swap Usage • CPU Usage • Network Packet Error Rate
Cubrid	• CAS Usage

클라우드 장애관리와 성능상의 특화기능

APM은 모니터링을 넘어 클라우드 환경에서 장애에 대한 원인규명과 분석을 할 수 있는 기능이 있어야 합니다.

고려사항

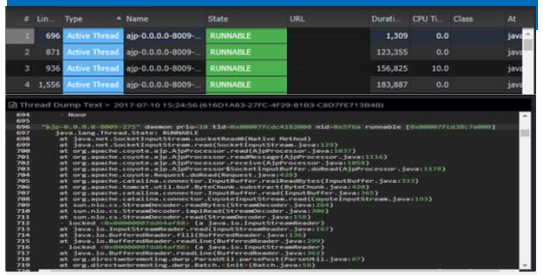
고려사항

- ⊘ 장애 발생 시 엔지니어가 도착하기 전에 장애 분석이 필요함
- ⊘ APM 사용시 APM 에서 제공하는 정보가 궁금할 때 즉각 문의 할 수 있어야 함

방안

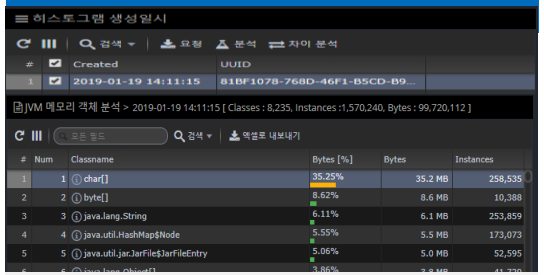
- 애플리케이션 성능 모니터링 및 진단, 장애 원인 분석을 통해
- 서비스를 최적의 상태로 운영할 수 있는 모니터링 솔루션이다.

Java 스레드 덤프 분석기



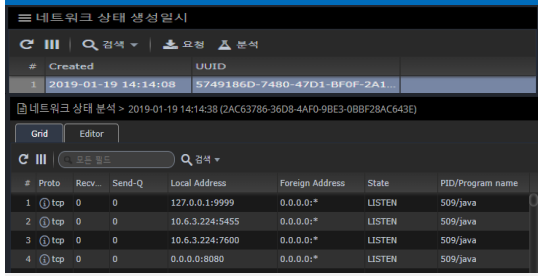
Lock을 추적가능, URL 정보표시

Java 메모리 누수 분석기



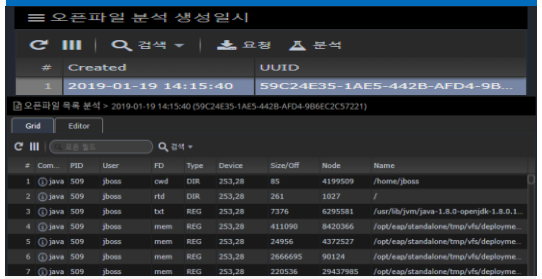
Java 메모리를 점유한 객체 분석/비교

네트워크 상태 분석기



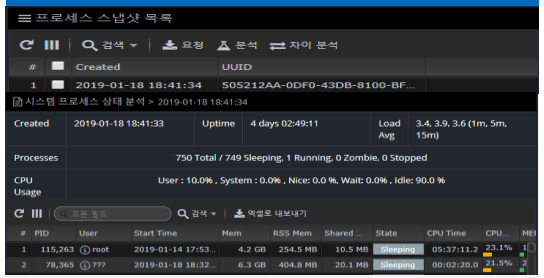
Java 프로세스,시스템이 사용중인 네트워크 분석

오픈파일 분석기



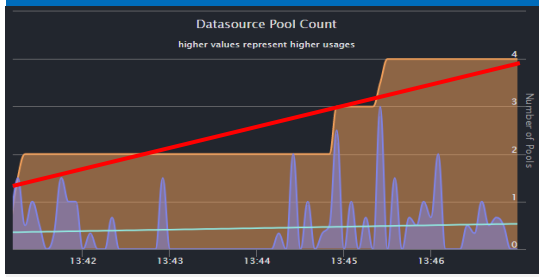
Java 프로세스가 오픈한 파일 분석

시스템 프로세스 분석기



시스템의 프로세스 CPU, 메모리 사용량 분석/비교

데이터 추세 분석



과거 데이터의 증감 추세를 분석하는 기능

사용자 관점의 주기적인 서비스 품질점검 서비스

서비스 접속이 안된다고 불편이 접수되는데, 내부에서는 인지하지 못하는 상황을 겪어보진 않으셨나요?

고려사항

- ⊘ 인터넷망 사용자의 접속이 문제가 있는 경우 메신저, SMS, 이메일 등으로 상황전파가 되어야함
- ⊘ 사용자 관점의 SLA 를 일간/주간/월간 등의 주기로 보고서를 생성해서 전달해야 합니다.

- 사용자 입장에서 주기적인 서비스 이상유무를 판단합니다. (예를 들어, 서버는 이상이 없는데 네트워크 장비 등의 문제로 서버 접속이 안되는 경우)
- 서비스별로 URL 기준으로 정의하며, 요청 성공율, 가용성, 평균응답시간, 호출 횟수 등의 서비스 정상 유무를 확인 합니다.

헬스체크 기능 활용

헬스체크 기능

- 서비스에 대한 주기적인 생사 확인
- 평균 응답 시간 추이 확인
- 특정 기능/페이지 상태 점검
- 재 배포 후 다수의 페이지에 대한 일괄 점검
- 사용자 관점의 서비스 상태 확인
- 인프라 구간별 서비스 상태 확인
- 복합적인 점검 수행과 오류에 대한 사용자 알림
- 스크립트를 이용하여 프로세스/파일/자원 사용량 등을 주기적으로 모니터링

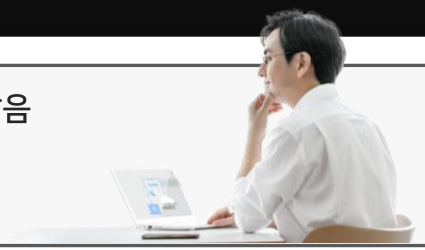
멀티 애플리케이션 운영 환경에서 클라우드 모니터링

여러 애플리케이션을 동시 운영 시 자원을 효율적으로 활용할 수 있을 까요?

고려사항

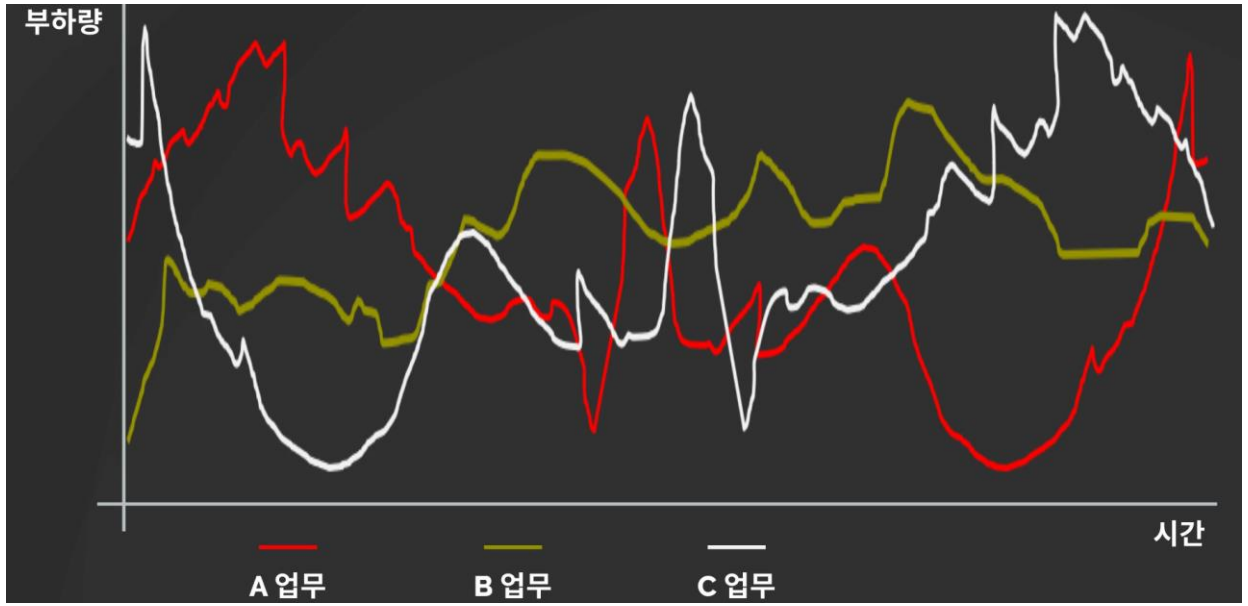
고려사항

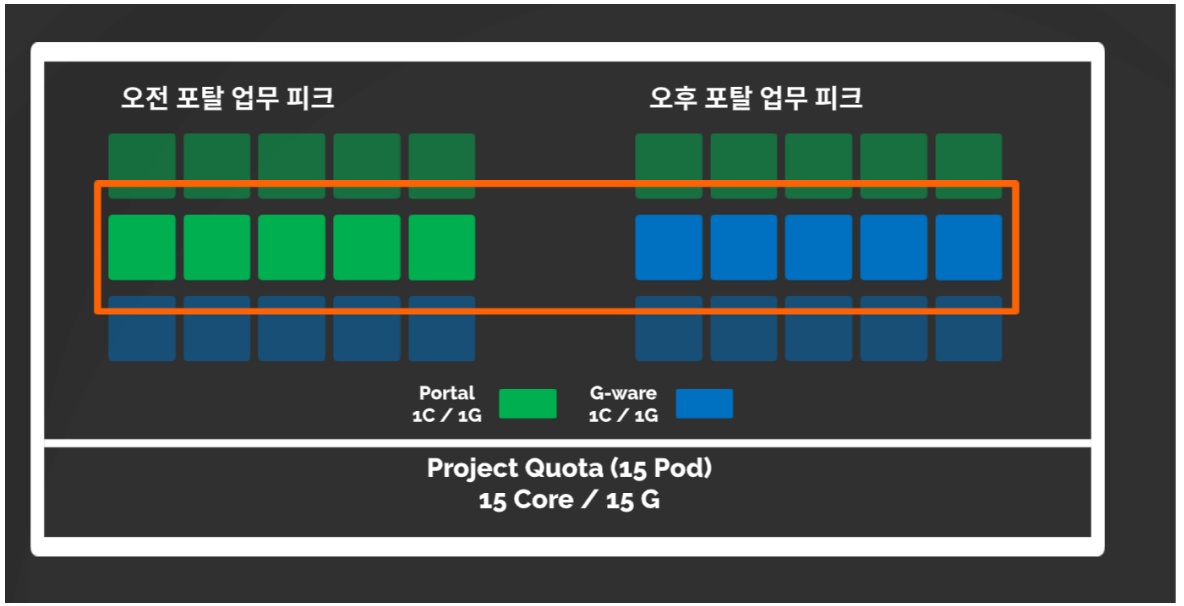
- ⊘ Peak 시간 대가 다른 여러 애플리케이션을 운영 시 기존의 시스템 구성과 사이징은 적합하지 않음
- ⊘ 관리자가 매일 시스템사항을 모니터링하고 그에 맞게 수작업으로 자원을 조정하기 어려움
- ⊘ 업무 별로 Peak 시 사용량 기준으로 사이징 하여 구축 비용에 대한 상당한 부담



방안

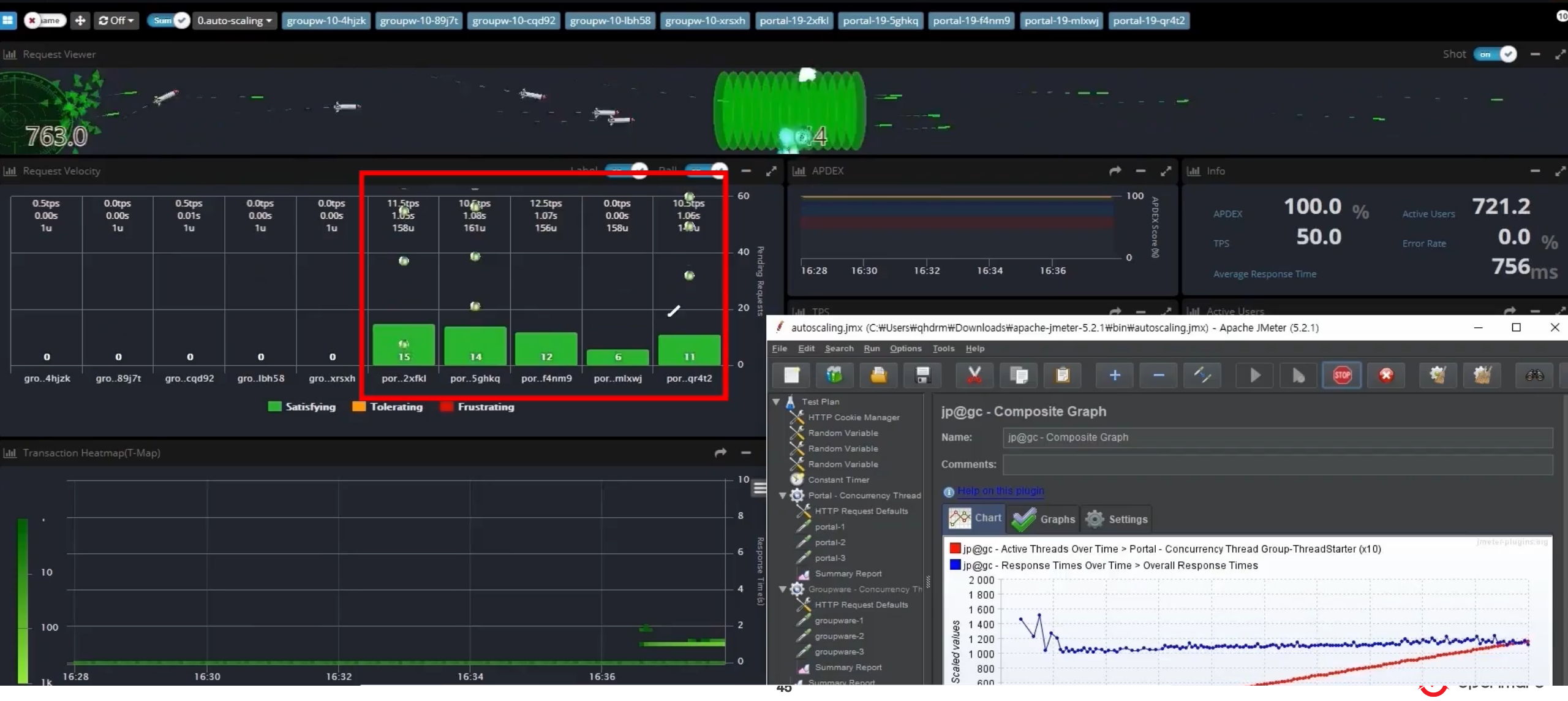
- 관리자의 개입없이도 시스템 부하량에 맞추어 자원일
- 클라우드 서비스 환경에서도 서비스 품질에 중요한 요소인 응답시간과 TPS 를 보장
- 자동확장에 의한 자원 효율성 확보를 통한 시스템 구축 비용 20% 이상 절감





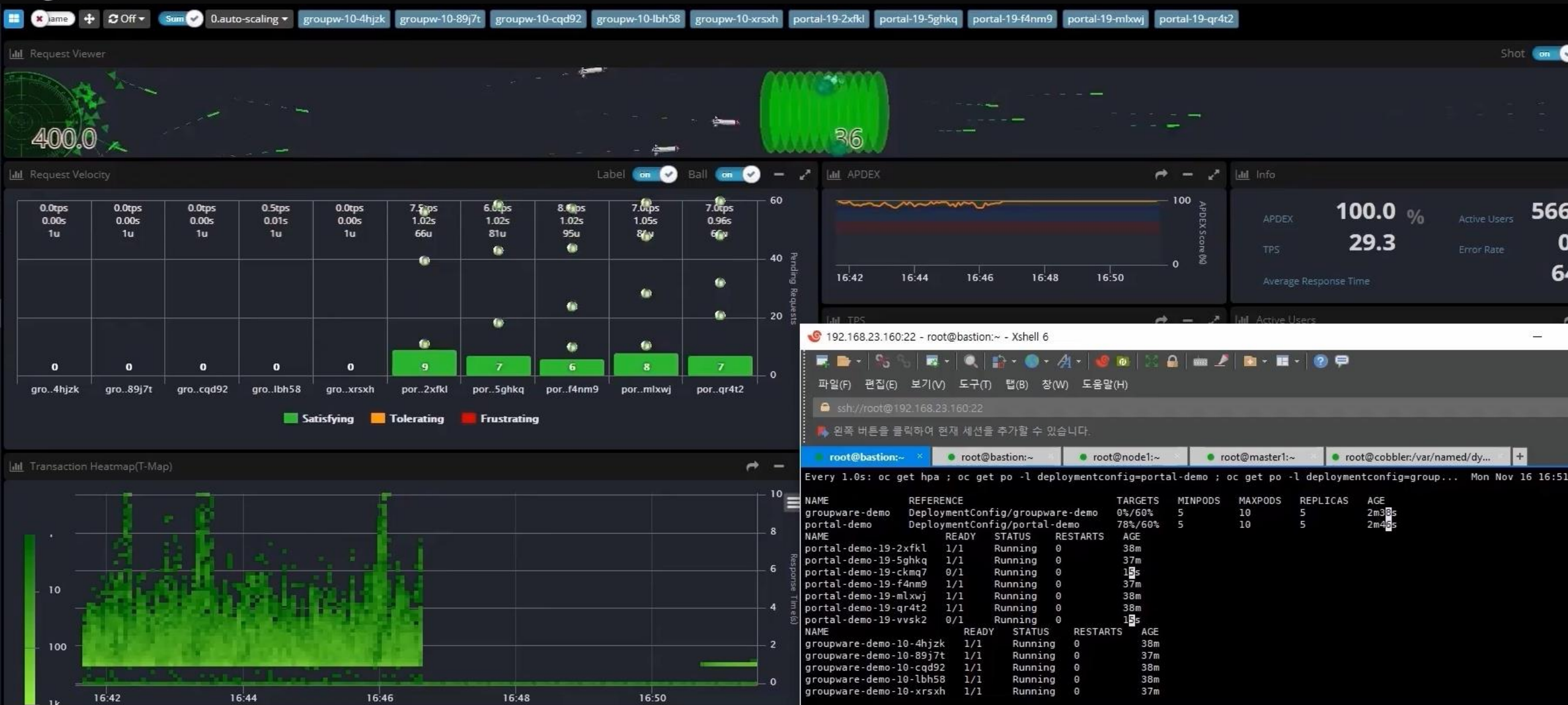
T-01 : 인스턴스 개수 한정된 환경(기존 나라통계)

기존에는 가상화 환경으로 부하 상황에서 **리소스를 효율적으로 활용할 수 없음**



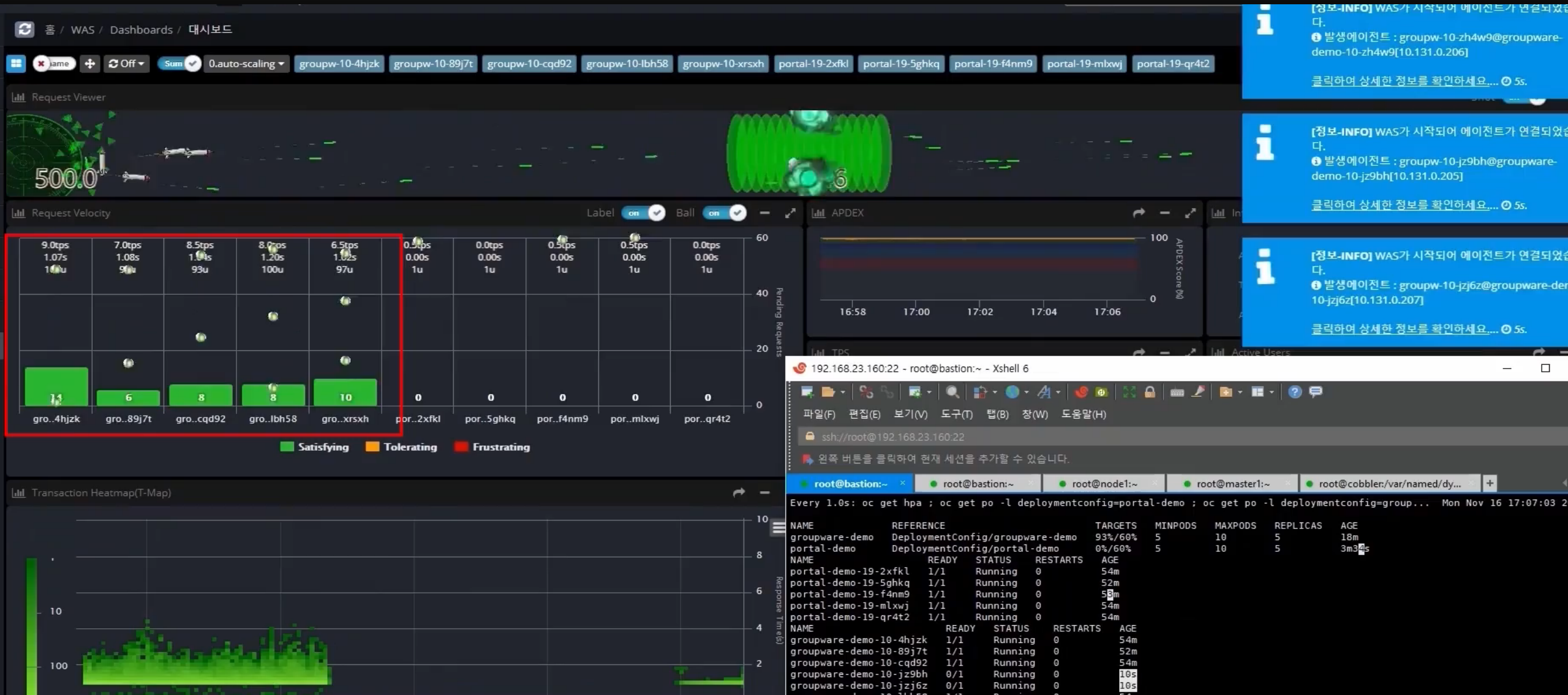
T-02 : 전국사업체조사 Peak일 때 자동부하분산 환경 부하테스트

부하에 따른 컨테이너 자동확장으로 응답시간 보장과 TPS 증가



T-03 : 초중고 사교육비조사 Peak일 때 자동부하분산 환경 부하테스트

부하에 따른 컨테이너 자동확장으로 응답시간 보장과 TPS 증가



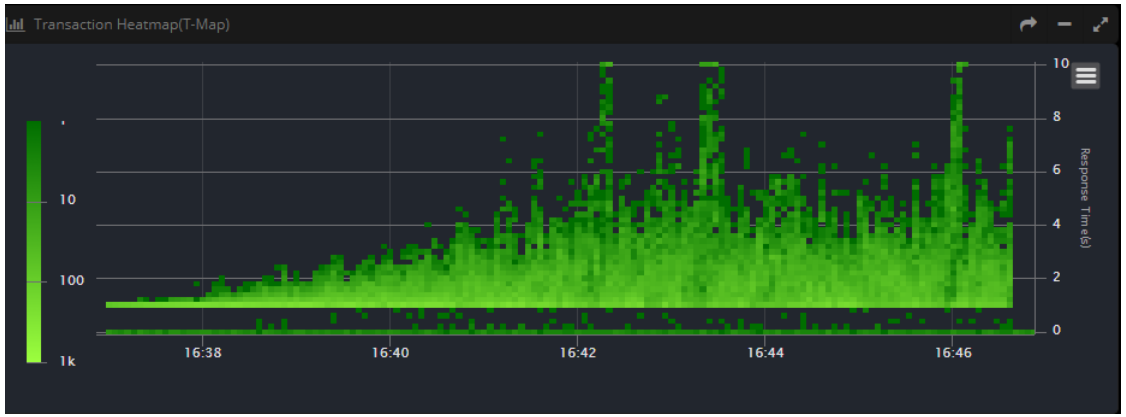
DEMO – 부하 테스트 결과 비교

Peak 시점이 다른 통계조사에 대해 별도 개입없이 자동자원할당이 가능함을 확인

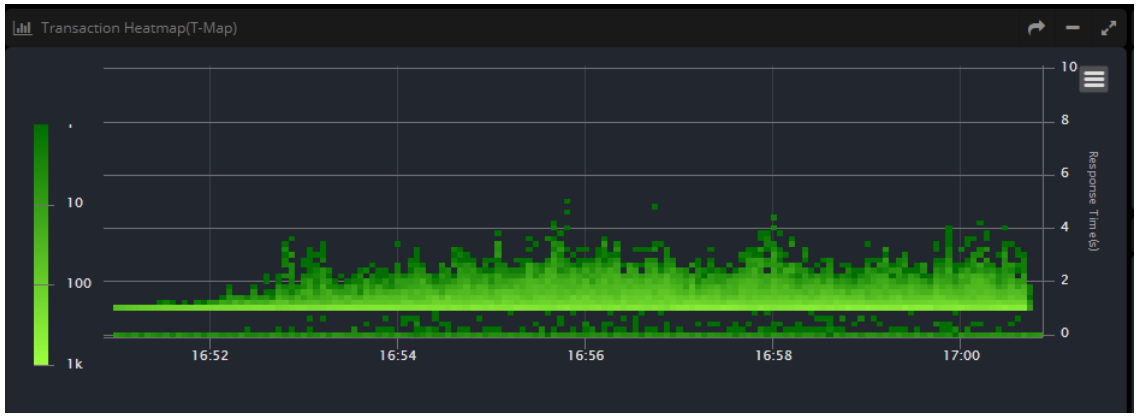
- 기존환경과 비교하여 자동자원할당이 되는 환경이 **1.7 배** 많은 양을 처리하며, 평균응답시간이 **2.5 배** 빠름

테스트 케이스	테스트 내용	시뮬레이션 테스트 환경	처리량	**처리량 비교**	평균응답시간	**응답시간비교**	최소응답시간	최대응답시간	TPS
T-01	자동자원할당이 안 되는 환경에서 부하테스트	기존 나라통계 환경	53,582	100%	2,374	100%	1,010	19,979	88
T-02	Peak 시점이 다른 자동 자원할당이 되는 환경에서 부하테스트	전국사업체 조사 Peak 일 때 환경	93,869	169%	919	258%	10	4,589	155
T-03	Peak 시점이 다른 자동자원할당이 되는 환경에서 부하테스트	초중고 사교육비 조사 Peak일 때 환경	91,394	164%	971	244%	10	5,134	151

T-01 자동자원할당 안되는 기존 환경 응답시간분포



T-02/03 자동자원할당이 되는 환경 응답시간분포

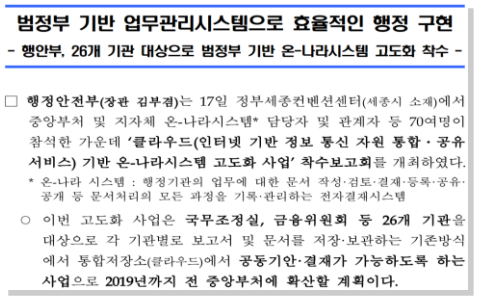


행정안전부 - 국가 80 여개 중앙부처 그룹웨어(온-나라) 시스템

한국지역정보개발원 온-나라 클라우드 문서 2.0 사업(2018 년) - Red Hat PaaS & OPENMARU APM

프로젝트 명	온-나라 클라우드 문서 2.0
얼마나 중요한 업무인가?	정부부처 20군데와 지방자치단체 약 80군데 등100만 공무원이 사용하는 문서 결재 시스템
규모는 어떻게 되는가?	Wokrer Node 80대 규모
오픈마루 역할은?	PaaS 전문인력 상주와 OPENMARU APM으로 성능관리를 담당하여 안정화 지원

☑ OPENMARU APM은 국내 최초, 최다 컨테이너 & 클라우드 환경을 지원



범정부 기반 업무관리시스템으로 효율적인 행정 구현

- 행안부, 26개 기관 대상으로 범정부 기반 온-나라시스템 고도화 착수 -

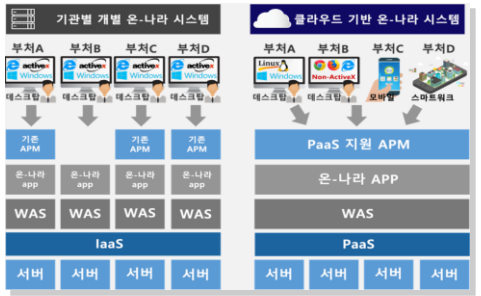
□ **행정안전부(장관 김부겸)**는 17일 정부세종컨벤션센터(세종시 소재)에서 중앙부처 및 지자체 온-나라시스템* 담당자 및 관계자 등 70여명이 참석한 가운데 **'클라우드(인터넷 기반 정보 통신 자원 통합 · 공유서비스) 기반 온-나라시스템 고도화 사업' 착수보고회**를 개최하였다.

* 온-나라 시스템 : 행정기관의 업무에 대한 문서 작성·검토·결재·등록·공유·공개 등 문서처리의 모든 과정을 기록·관리하는 전자결재시스템

○ 이번 고도화 사업은 **국무조정실, 금융위원회 등 26개 기관**을 대상으로 각 기관별로 보고서 및 문서를 저장·보관하는 기존방식에서 통합저장소(클라우드)에서 **공동기안·결재가 가능하도록 하는 사업**으로 **2019년까지 전 중앙부처에 확산할 계획**이다.

행정안전부 보도 자료
범정부 기반 온-나라 시스템 고도화 착수

클라우드 전환 시 노후 장비 교체 대비
2.6배의 비용 효과 발행

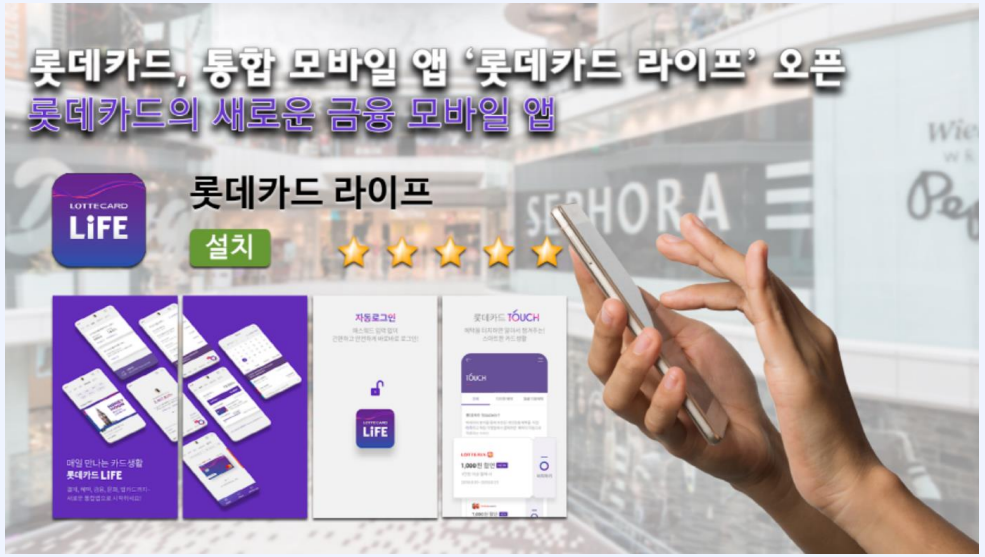


클라우드 온-나라 문서 2.0
개념도

온나라 클라우드 APM
성능 테스트

롯데카드 – LP (Life Platform)

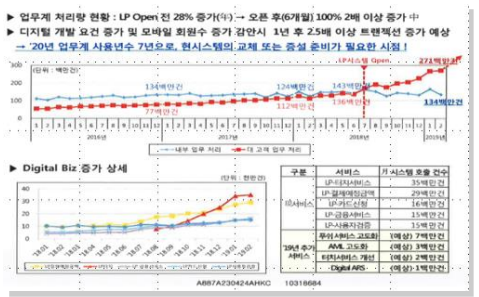
롯데카드의 계정/정보/채널계를 모두 Digitalization 인프라 클라우드 플랫폼을 구현했습니다.



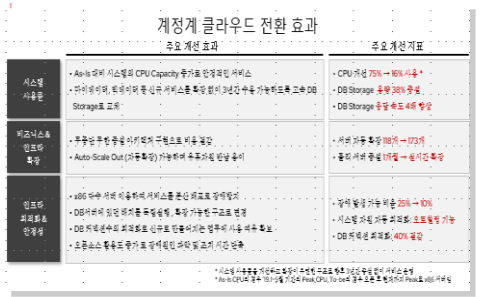
프로젝트 명	롯데카드 채널계 클라우드 구축
얼마나 중요한 업무인가?	금융권 최초로 채널계 시스템 도입으로 대고객 접점 서비스
규모는 어떻게 되는가?	채널계 전체 클라우드 도입
오픈마루 역할은?	자동 확장, 축소되는 클라우드 컨테이너를 국내 최초로 OPENMARU APM으로 안정화 지원 및 성능, 튜닝

롯데카드의 클라우드 환경 도입 효과

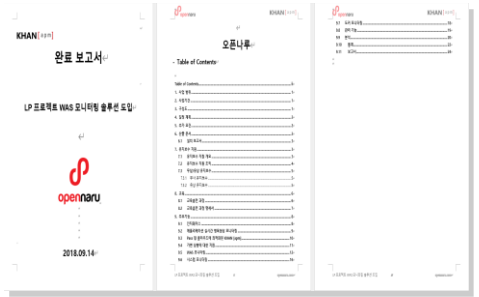
롯데카드, '레드햇 포럼 서울 2018' 최고 디지털 전환 상 수상



채널계 클라우드 도입 효과



계정계 클라우드 전환 효과



롯데카드 LP 프로젝트 OPENMARU APM 도입 완료 보고서

한국교육방송공사 - 코로나 대응을 위한 온라인 클래스 시스템

EBS 온라인 클래스에 도입되어 동시접속자 130만명 , VM 600 대 , WAS 인스턴스 1200 여 개 모니터링

실제 600대 WEB/WAS를
OPENMARU APM으로 모니터링하는 Dash Board 화면

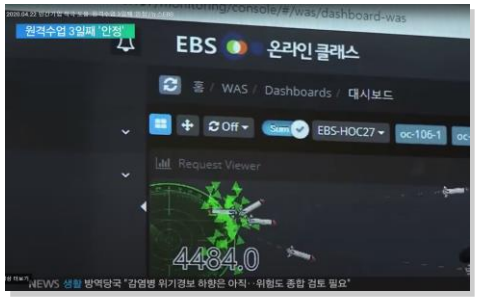
프로젝트 명	EBS 온라인 클래스
얼마나 중요한 업무인가?	코로나로 인하여 초,중,고 300백만명의 학생이 접속하는 온라인 클래스 구축 업무
규모는 어떻게 되는가?	WEB/WAS 600대 / 인스턴스1,200개 규모
오픈마루 역할은?	1주일 내 600대 머신에 1,200개 인스턴스 구축 및 APM으로 오픈지원, 성능 튜닝 및 안정화 지원

☑ OPENMARU APM으로 안정화를 실현한 실제 EBS 방송 화면

교육 차관에게 OPENMARU APM으로
보고하는 화면

600여대의 서버의 총 1,228개 WAS 모니터링

실제 인스턴스 1,200개 그룹대시보드
모니터링 화면 -2



실시간 Active User를
확인할 수 있는 모니터링 화면


openmaru