세원아이티를 위한 클라우드 네이티브 무상 컨설팅
찾아가는 클라우드 네이티브 세미나

찾아가는 클라우드 네이티브 세미나

openmaru
오픈마루㈜

1 CentOS의 EOS와 OpenShift로의 전환

- CentOS To RHEL vs OpenShift 비교

2 클라우드 네이티브 사전 질의 내용

- 공공기관의 클라우드 네이티브 적용 범위
- 전환시 고려사항
- 보안 해결 방안
- 비용 산정 고려 사항
- 도입 사례

클라우드 네이티브 무상 컨설팅

찾아가는 클라우드 네이티브 세미나

CentOS의 EOS와 OpenShift로의 전환

CentOS의 EOS와 ISMS

openmaru
오픈마루(주)

CentOS의 EOS

1. CentOS Linux 8의 업데이트 및 릴리스는 2021년 12월 31일에 중단
   - CentOS는 향후 CentOS Stream 으로 전환
2. CentOS Stream는 RHEL 업스트림 (개발) 배포판
   - CentOS 8사용자는 앞으로 CentOS Stream로 전환하거나 운영 환경에서 사용한다면 RHEL로 전환
3. CentOS Linux 7에 대해서도 2024년 6월 30일에 중단

ISMS

- 2.10 시스템 및 서비스 보안관리
  - 2.10.8 패치관리
    소프트웨어, 운영체제, 보안시스템 등의 취약점으로 인한 침해사고를 예방하기 위하여 최신 패치를 적용하여야 한다. 다만 서비스 영향을 검토하여 최신 패치 적용이 어려울 경우 별도의 보완대책을 마련하여 이행하여야 한다.
- ISMS 인증을 준비중이거나 심사중일 경우 OS의 패치 관리는 매우 중요한 이슈
- EOL된 OS를 사용하는 경우 인증 심사에서 대처 미흡으로 중결함 판정 받는 사례 발생

CentOS의 대안으로 또 오픈소스 OS ?


세일즈포스, 센트OS에서 레드햇으로 인프라 마이그레이션

박시현 기자 | 승인 2023.09.20 16:43 | 댓글 0

레드햇 엔터프라이즈 리눅스로 글로벌 하이브리드 인프라 표준화

Red Hat

[디지털경제뉴스 박시현 기자] 레드햇은 세일즈포스가 레드햇 엔터프라이즈 리눅스로 글로벌 하이브리드 인프라를 표준화한다고 발표했다.

http://www.denews.co.kr/news/articleView.html?idxno=27342

- 6년전 세일즈 포스는 레드햇 리눅스에서 CentOS 7로 마이그레이션 수행
- 최근 **CentOS 7의 EOS로 20만대의 서버들을 RHEL 9로 전환 수행**
- **세일즈포스는 '레드햇 엔터프라이즈 리눅스'**를 기반으로 하이브리드 클라우드 인프라의 전체 범위를 **표준화**
- **클라우드 규모의 운영체계를 관리하는 사소한 일에 얽매이지 않고 고객 가치 제공에 집중**

- **알마리눅스와 로키 리눅스는 현재 어떻게 호환성을 보장**하는가? 이 약속을 지킬 것임을 어떻게 보장하는가?
- 알마리눅스와 로키 리눅스 조직에 비즈니스 및 미션 크리티컬 시스템의 **적절한 패치, 업데이트 및 유지보수에 필요한 기술 전문성**이 있는가?
- OS는 기업에서 실행하는 모든 애플리케이션, 데이터베이스 등의 기반
- OS는 **소소한 비용을 절약하기 위해 클론을 선택하기에는 전혀 적합하지 않은** 부분


글로벌 칼럼 | 센트OS 대안으로 또 다른 RHEL 클론을 선택해선 안 되는 이유

Red Hat

https://www.itworld.co.kr/news/298057

CentOS EOS의 대응 방안, CentOS To RHEL? CentOS To OpenShift?

openmaru
오픈마루(주)

1. CentOS To RHEL

- 운영중인 CentOS의 Version을 기술지원이 포함된 버전으로 업그레이드하고 Convert2RHEL을 이용하여 레드햇 리눅스로 변환하는 방식
- Convert2RHEL이란 오라클 리눅스, CentOS, 알마리눅스, 록키리눅스를 레드햇 리눅스로 변환하기 위해 레드햇이 제공하는 변환 도구

2. CentOS To OpenShift

- CentOS로 운영중인 레거시 환경(베어메탈,가상화)을 클라우드 네이티브 기반 환경인 Red Hat OpenShift Container Platform으로 마이그레이션하는 방안
- 레드햇 리눅스로 업그레이드시에도 레거시 환경의 단점들은 존재, 변환이 아닌 마이그레이션을 진행하여 클라우드 네이티브의 장점을 누릴 수 있는 방안
- RHMTA(Red Hat Migration Toolkit for Application)를 활용하여 마이그레이션 난이도 측정
- 애플리케이션 단위로 컨테이너라이즈 필요

왜 CentOS To OpenShift인가?

openmaru 오픈마루㈜

복잡성

레거시 환경의 CentOS 업그레이드시 최소 2번 이상의 업그레이드 및 변환 작업 필요

Existing Installation Type	Post-conversion Installation
CentOS 7 Linux	Red Hat Enterprise Linux 7
CentOS 8 Linux	Red Hat Enterprise Linux 8

CentOS 7의 경우 8로 변환 불가. 8로 업그레이드 후 변환 가능

클라우드 네이티브 도입

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

반복성

OS 및 소프트웨어의 업그레이드는 보안패치, 취약점 조치에 따라 반복하게 됨.

- Human Error 발생
- 서비스 중단 발생
- 관리자 개입 반복

자동화 → Red Hat OpenShift

라이선스 비용 절감

PaaS에 포함된 OS, Runtime, Middleware뿐만 아니라 레거시 환경의 보안소프트웨어 종속 제거

Red Hat

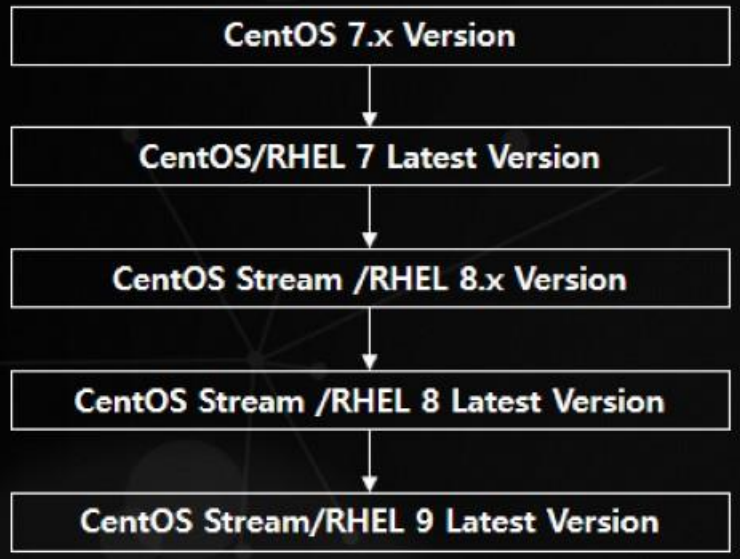
JBoss by Red Hat

EOS 대응을 위한 OS 업그레이드 소요시간

openmaru 오픈마루㈜

CentOS To Red Hat Enterprise Linux

- 최신 마이너버전이 아닐 경우 최신 메이저 버전으로 업그레이드 불가
- VM 및 Baremetal의 수량에 따라 작업시간 증가

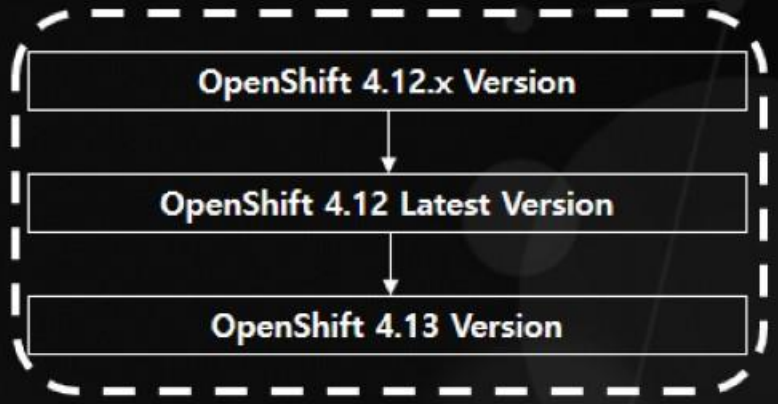


X VM or Baremetal Count

서버 1대에 약 2시간 소요 → **100대 VM: 400시간**

CentOS To OpenShift Container Platform

- OpenShift의 경우 OpenShift 버전 업그레이드시 Host OS(RHCOS) 자동 업그레이드 진행
- 클러스터 단위로 업그레이 진행



X OpenShift Cluster Count

Worker Node 6대 → **1개 클러스터: 7시간**

클라우드 네이티브 도입시 OS 업그레이드 절차 비교

openmaru
오픈마루㈜

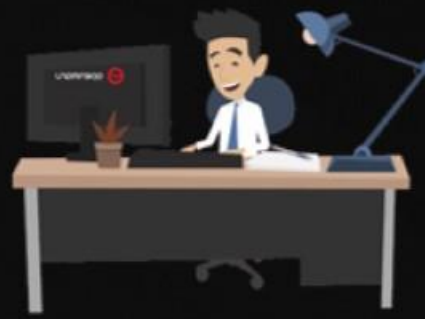
OpenShift 환경의 자동화된 OS 업그레이드

- 기존 환경과 달리 **컨테이너 기반의 애플리케이션이기 때문에 의존성 최소화**
- 업그레이드는 **관리자 개입 없는 OpenShift 자동 수행**
- **Red Hat에서 Upgrade Path 제공**

Legacy

- 5. 서비스 확인
- 4. 업그레이드 수행
  - RPM/Kernel Upgrade
  - OS 튜닝
- 3. Repository 설정
  - Minor Upgrade Repo
  - Major Upgrade Repo
- 2. 서비스 중지
- 1. 의존성 조사
  - Runtime 버전
  - 3rd party 버전
  - OS 튜닝 등

OpenShift



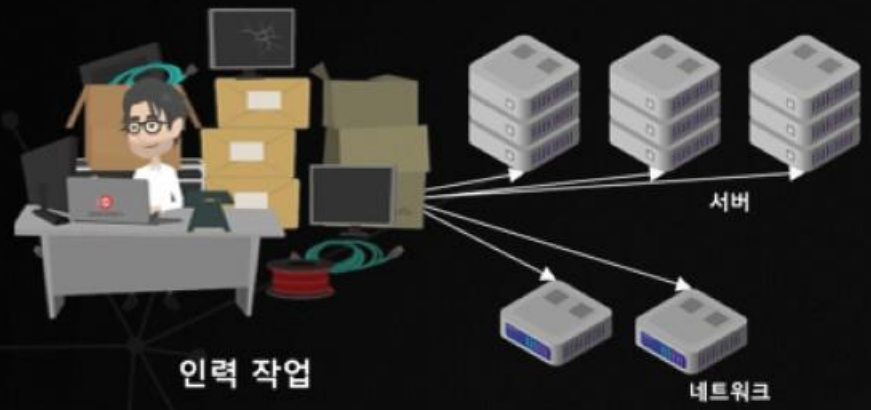
- 3. 서비스 확인
- 2. 업그레이드 수행
- 1. Registry 설정

Upgrade 자동화

OpenShift Container Platform을 통한 업그레이드 자동화

- **OpenShift Container Platform의 경우 버전 업그레이드시 Host OS(CoreOS)도 같이 업그레이드**
- **업그레이드 동작시 관리자 개입이 필요 없는 완전 자동화**

Before OpenShift



- 시스템 운영을 위한 관리 작업 증가
- 현장 작업 시간 증가
- Human Error 증가

After OpenShift

- 운영 기술 표준화를 통한 준비 시간 및 작업 시간 감소
- 시스템 일괄 설정 작업 시간 단축
- 시스템을 통한 작업으로 Human Error감소

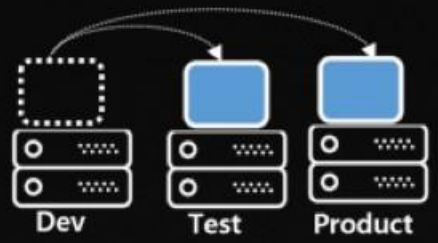
클라우드 네이티브 도입 효과

openmaru
오픈마루(주)

- 클라우드 네이티브 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

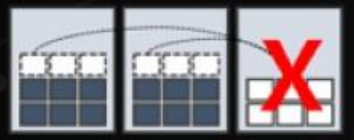


Consistency & Continuous

- 이미지 기반으로 구성, 배포 효율화
  개발과 운영 환경의 일관성

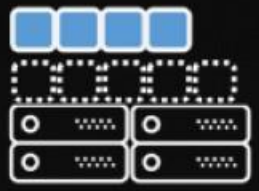
Rolling Update

- 업그레이드 또는 패치 시
  다운 타임은 제로 또는 최소화



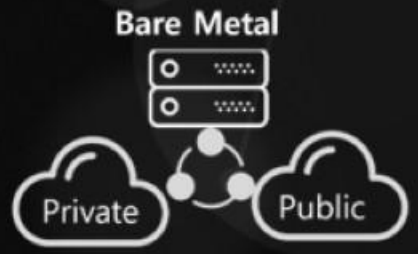
Self Recovery

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



Application Scaling

- VM 단위가 아닌 어플리케이션
  단위의 오토스케일링

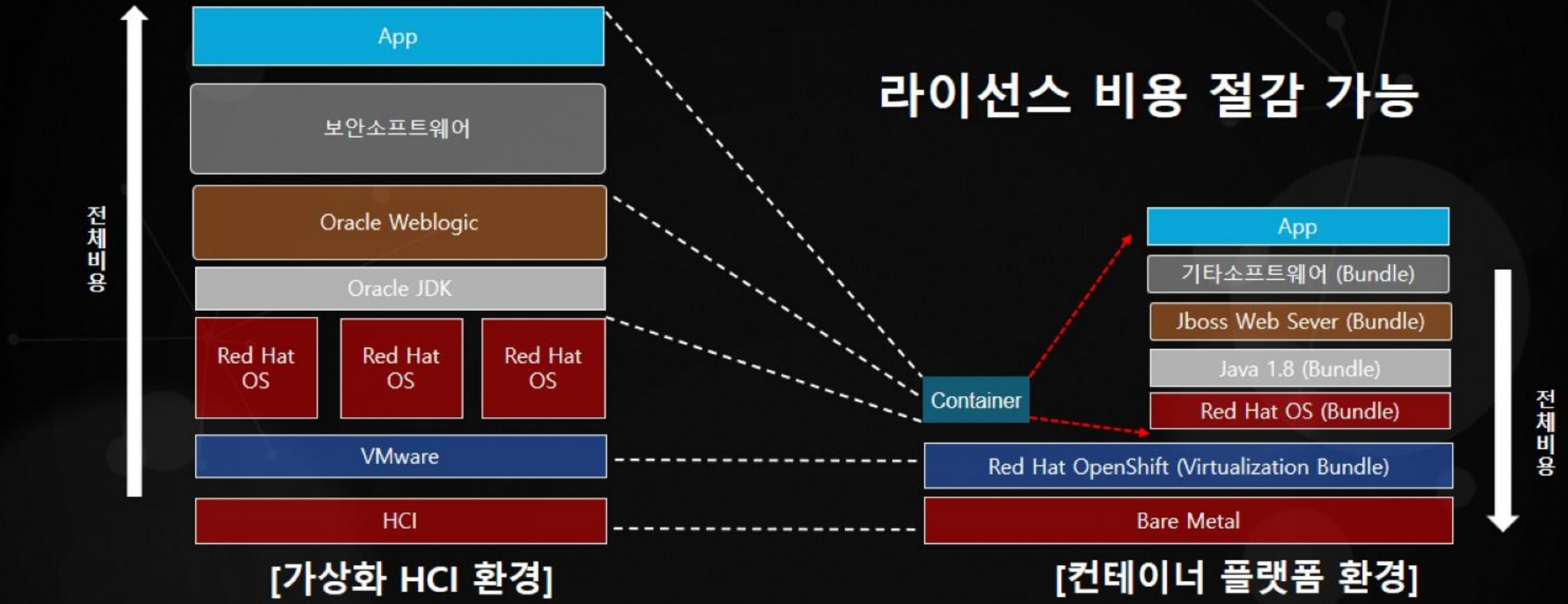


Portable

- 멀티/하이브리드 클라우드 기반
  어플리케이션/서비스 운영

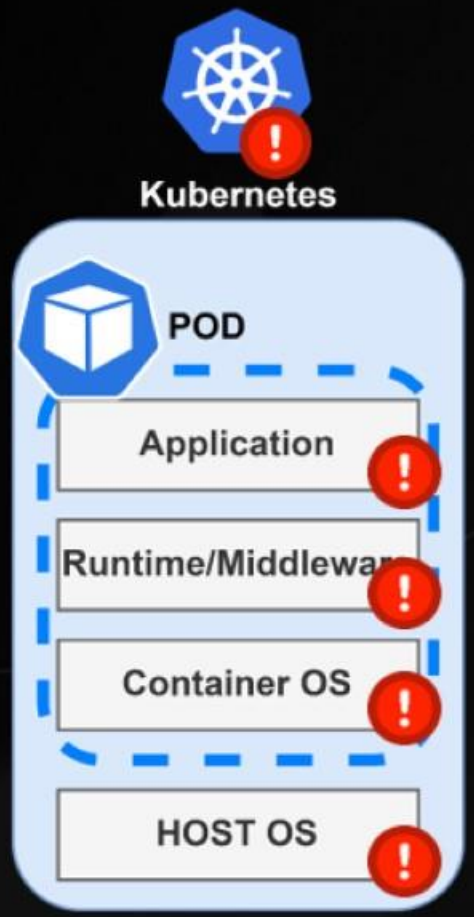
라이선스 비용 절감

- 컨테이너 환경에서의 가상화소프트웨어 / WAS / 보안소프트웨어 비용 제거



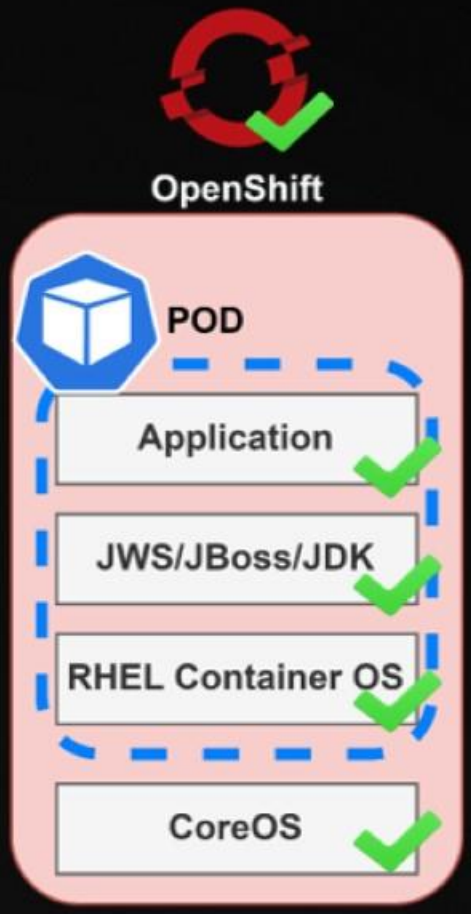
신뢰할 수 있는 Software 스택 제공

openmaru
오픈마루㈜



Kubernetes + DIY Stack

- 신뢰할 수 없는 컨테이너 이미지
  - 컨테이너 보안 업데이트 X
- QA팀을 통한 안정화 테스트 X
- HOST OS 유지보수
  - 업그레이드
  - 패치
  - 트러블슈팅
  - 구매비용 발생
- Runtime/Middleware 유지보수
  - 업그레이드
  - 패치
  - 트러블슈팅
  - 구매비용 발생



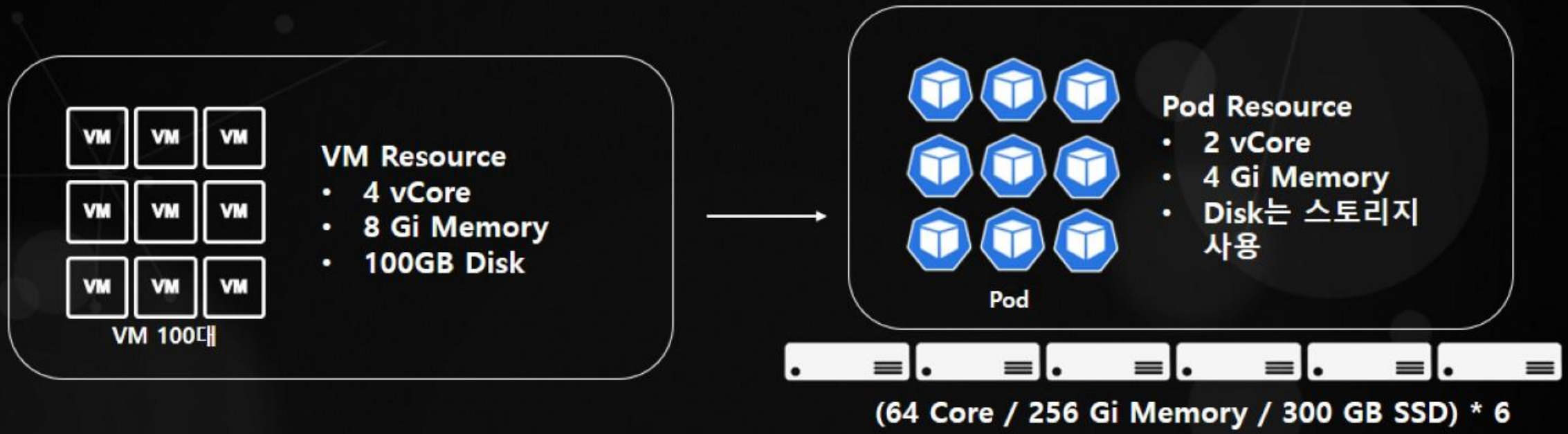
OpenShift Container Platform

- 신뢰할 수 있는 컨테이너 이미지
  - Quay, red hat registry
- QA팀을 통한 안정화 테스트
- Red Hat Core OS
  - 업그레이드 지원
  - 버그 패치 지원
  - 트러블슈팅 지원
  - 컨테이너 특화 OS
  - 구매비용 발생 X
- OpenJDK / JBoss Web Server, EAP
  - 업그레이드 지원
  - 패치 지원
  - 트러블슈팅 지원
  - 구매비용 발생 X
    (EAP의 경우 구매비용 발생)

VM 100대 기준으로 OpenShift Sizing

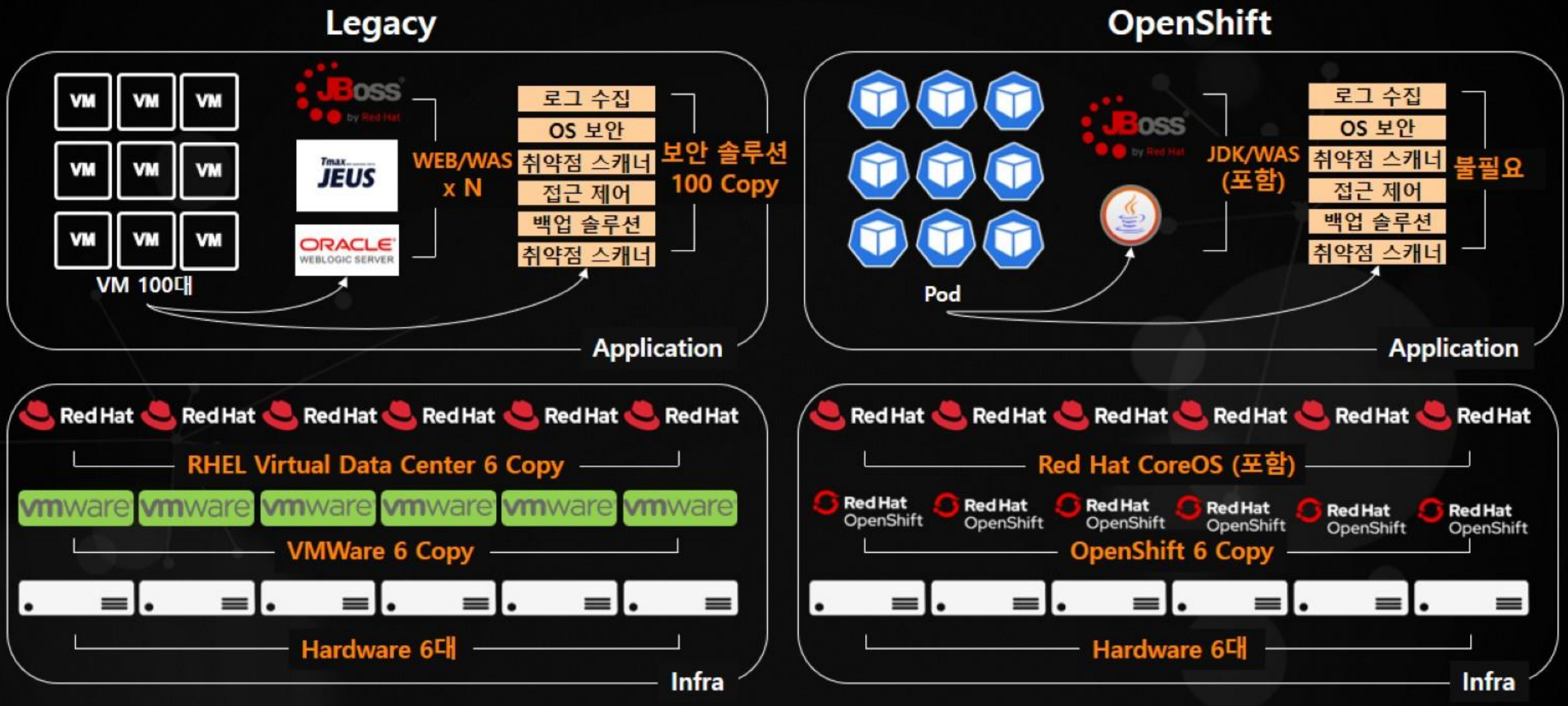
- VM 100대 시스템이 OpenShift로 변환하였을 때의 Sizing
  - VM의 리소스 산정은 4vCore / 8Gi Memory를 기준으로 산정

- OpenShift의 라이선스의 경우 2 Socket 기준으로 1 Copy 발생 → 64 Core : 128 vCore
  - 6대의 OpenShift의 Worker Node : 768 vCore 사용
  - 자동 자원 확장 / 노드 장애 상황 등을 가정하여 여유 자원 산정



레거시 환경과 클라우드 네이티브 환경 라이선스 수량 비교

openmaru 오픈마루㈜



아직도 WAS BMT 나 POC를 하시나요?

openmaru 오픈마루㈜

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

WAS 제품 BMT

ORACLE WEBLOGIC

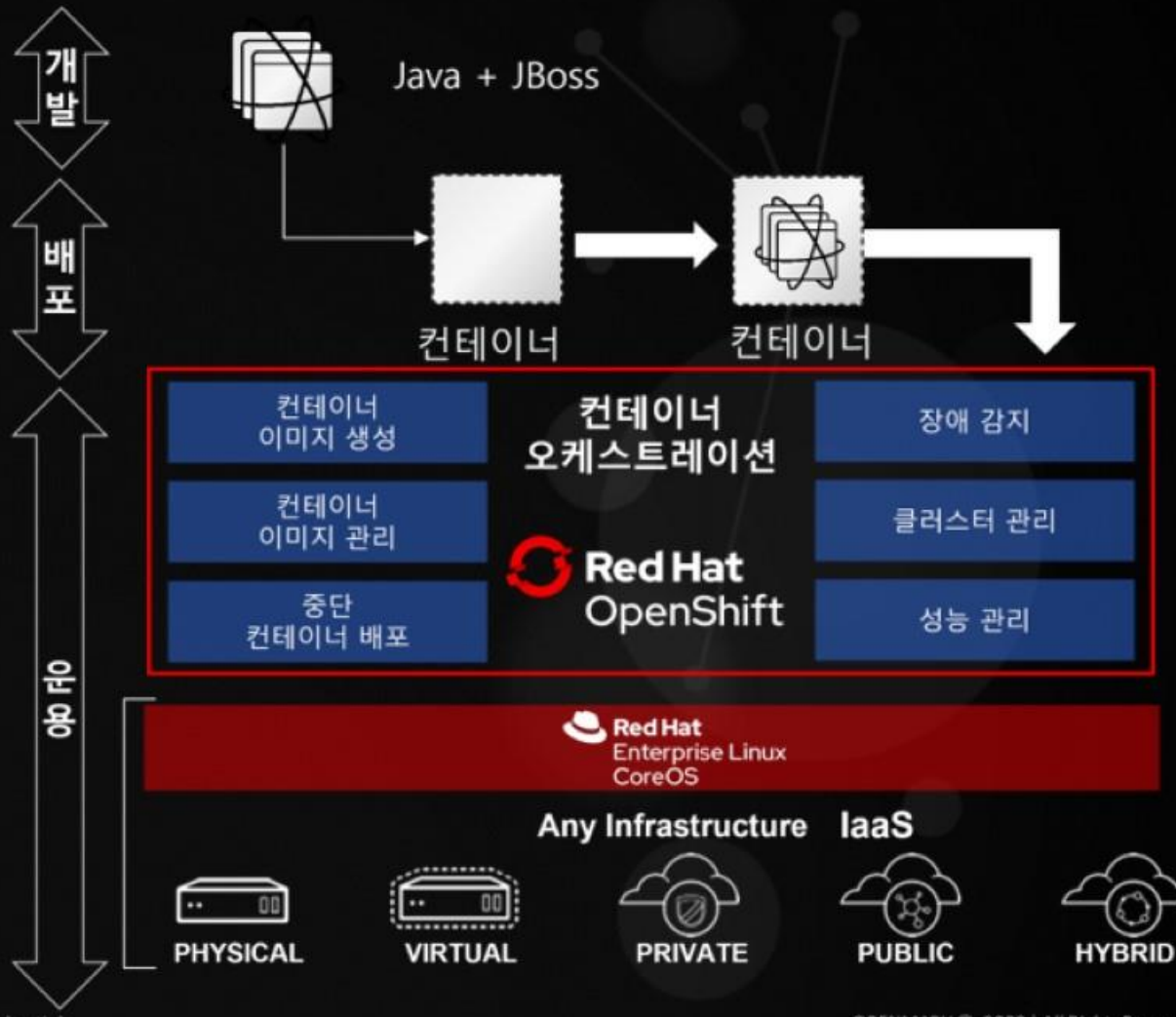
JBoss by Red Hat

WebSphere

Tmax 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



레거시 환경에서 필요한 서버 보안

- OS 보안, 서버 접근제어 등 보안 5종 S/W를 OS 설치
  - OS 마다 설치하기 때문에 물리서버는 1Copy이고, 가상화는 VM 개수 만큼 설치
- 서비스에 따라 부가적인 보안 소프트웨어 필요(개인정보 검출, 비정형 암호화 등)
- 법적인 근거로 인해 서버보안 – 전자금융법, ISMS 등의 요건

컨테이너 환경에서 필요한 보안

openmaru
오픈마루(주)

왜 AS-IS 환경의 보안들이 필요하지 않을까?

실질적으로 컨테이너환경에서 필요한 보안들

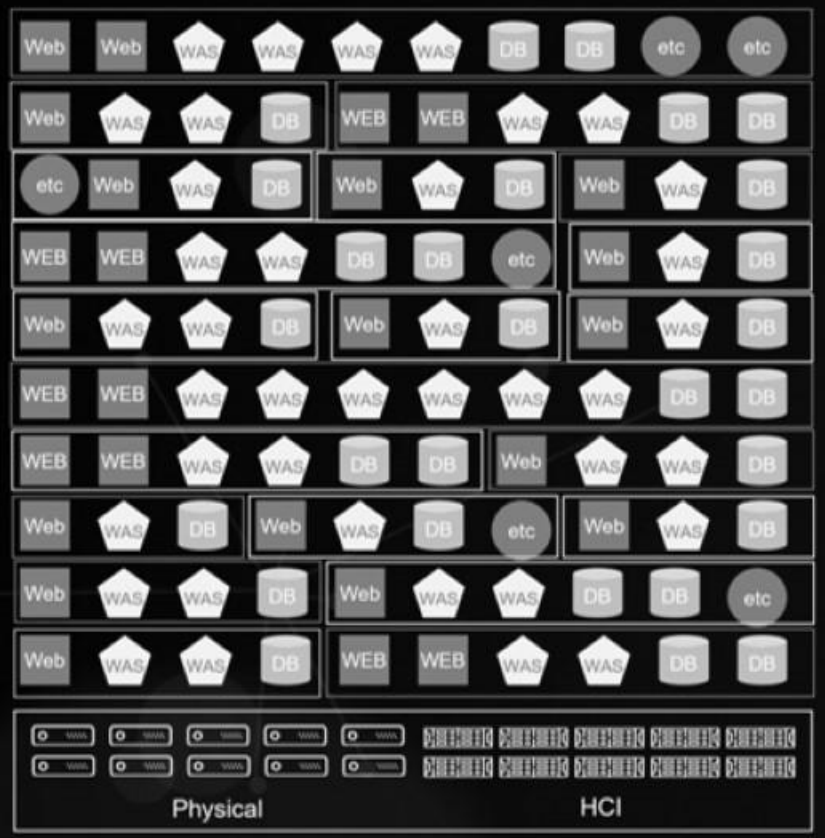
- 컨테이너 실행 권한에 대한 보안 : SCC
  - Container breakout
- 취약한 Container Image 사용
  - 악성코드, 채굴 프로그램이 포함된 이미지
- Role Base Access Controle : RBAC
  - 사용자별 필요 권한 부여
- 컨테이너 플랫폼의 Audit 로깅
  - EFK
- 신뢰할 수 있는 컨테이너 런타임 보안
  - Capabilities, SELinux, Seccomp & Namespace
- 컨테이너간의 네트워크 격리
  - Network Policy

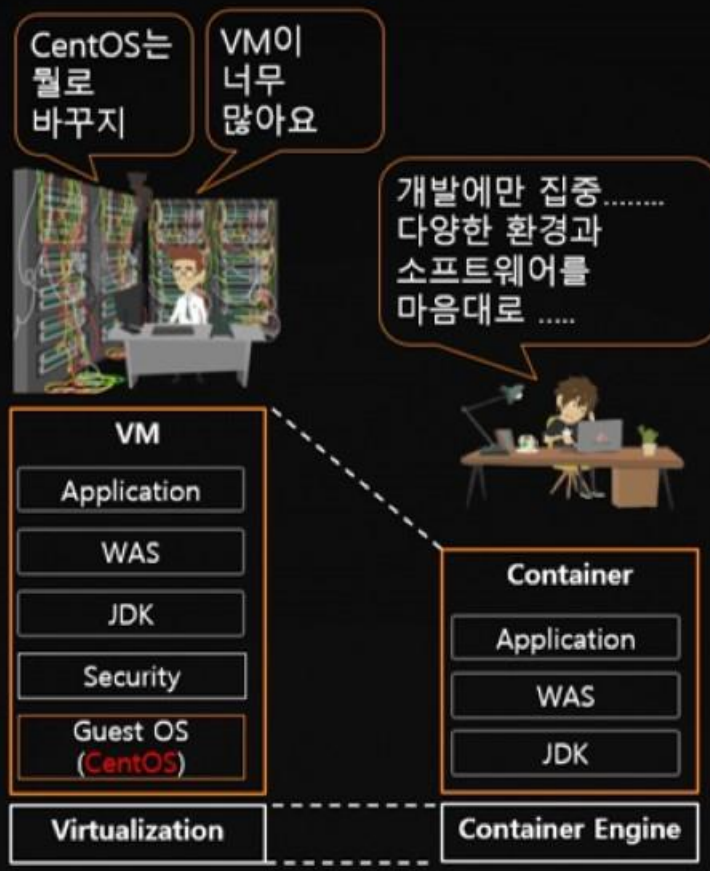
필수 보안 프로그램 설치 방안

CentOS To OpenShift시 구성 변화

openmaru
오픈마루㈜

시스템 마다 Web/WAS/DB 형태로 구축





- 물리서버나 가상화 형태로 WEB/WAS/DB 3 티어로 구축
  - Guest OS 에 따른 OS 라이선스 이슈와 보안 솔루션 구매 비용 발생

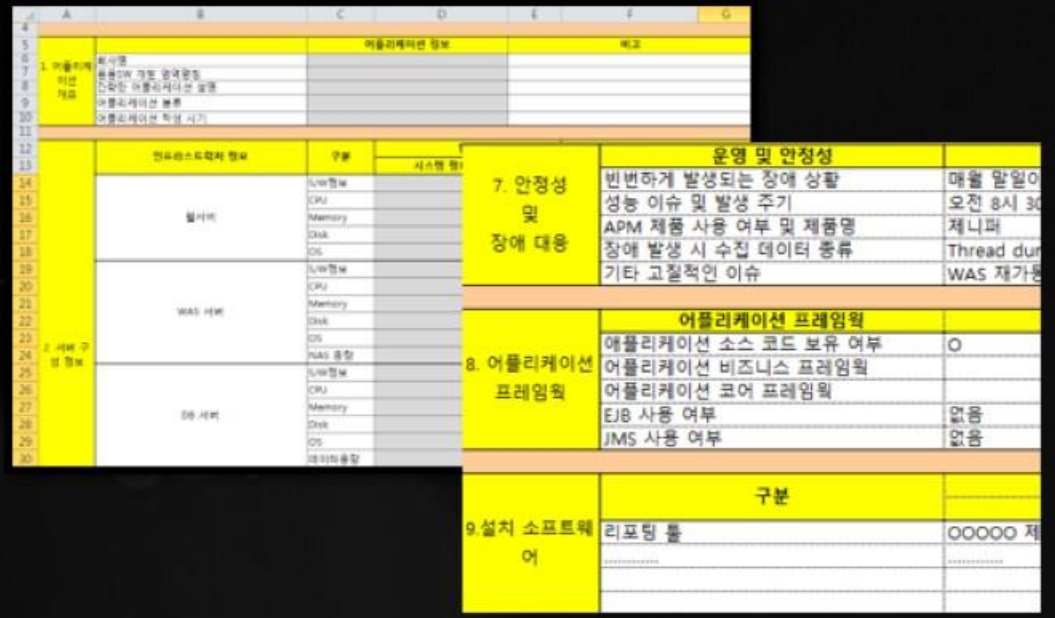
S/W 를 플랫폼에서 제공 – 개발에만 집중

- 애플리케이션에 필요한 S/W 만 컨테이너로 배포
  - 클라우드 네이티브로 전환, DevOps, CD/CD 기반

CentOS To OpenShift 난이도 분석 방안

애플리케이션의 마이그레이션이 가장 중요 !

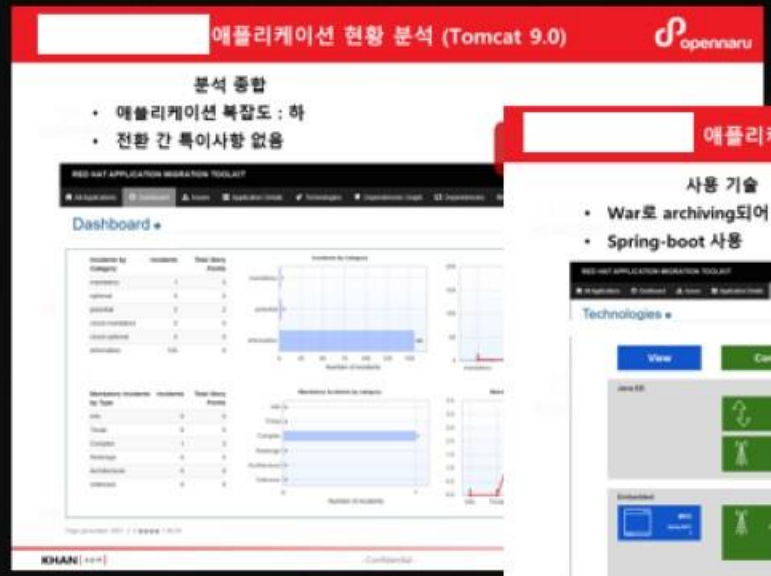
- 애플리케이션의 컨테이너화가 가능 유무, 난이도, 필요 공수 등을 분석이 반드시 필요.
- 일부 시스템에 종속되는 애플리케이션의 경우 마이그레이션이 불가능할 수 있음.
- 별도의 진단 도구를 활용한 분석 수행



7. 안정성 및 장애 대응	운영 및 안정성	
	빈번하게 발생되는 장애 상황	매월 말일
	성능 이슈 및 발생 주기	오전 8시 3
	APM 제품 사용 여부 및 제품명	제니퍼
	장애 발생 시 수집 데이터 종류	Thread du
	기타 고질적인 이슈	WAS 재가

8. 어플리케이션 프레임워크	어플리케이션 프레임워크	
	애플리케이션 소스 코드 보유 여부	O
	어플리케이션 비즈니스 프레임워크	
	어플리케이션 코어 프레임워크	
	EJB 사용 여부	없음
	JMS 사용 여부	없음

9.설치 소프트웨어	구분	
	리프팅 툴	OOOOO 제
	.....	.....

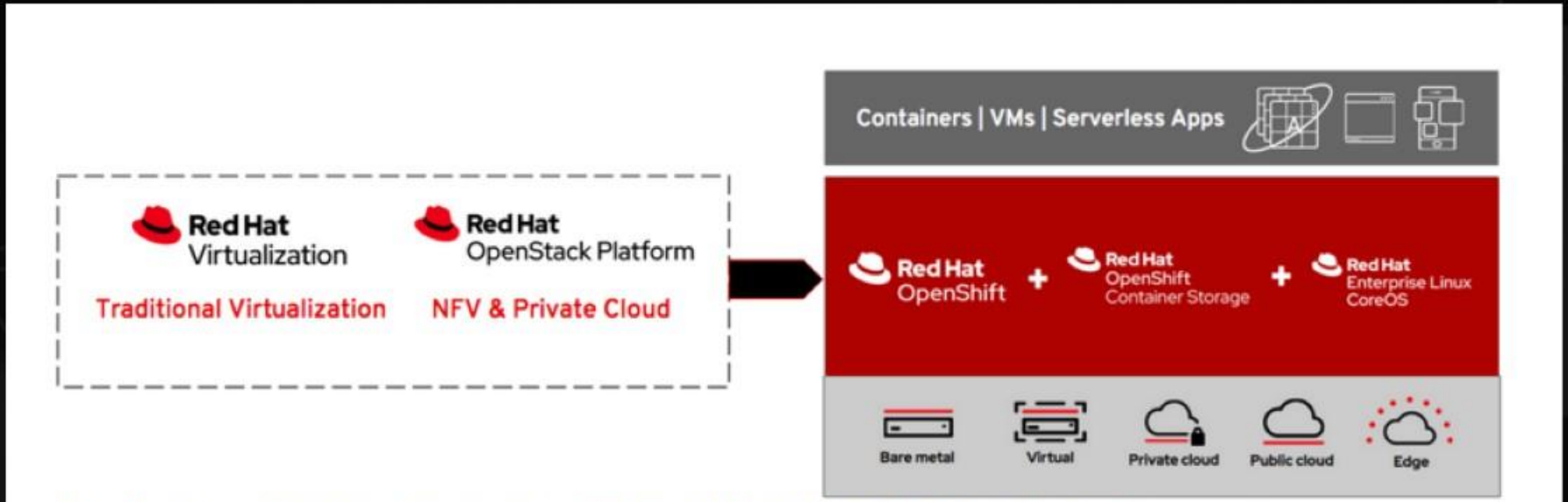


컨테이너화 불가능한 애플리케이션

openmaru
오픈마루㈜

OpenShift Virtualization 활용 방안

- OpenShift Virtualization는 가상 머신 생성/관리를 지원한s Operator
- 컨테이너 기반 애플리케이션과 동일한 플랫폼에서 VM 기반 워크로드를 운영
- Windows 혹은 별도 솔루션 같은 Containerize가 어려운 자원들에 대한 지속 운영 가능
- 추가 라이선스 금액 발생 X → Guest OS가 레드햇 리눅스일 경우에도 발생 X



레드햇이 OpenShift Virtualization으로 제공할 가상화의 미래, 현재 전통적인 가상화 integration
https://cloud.redhat.com/blog/blog-openshift-virtualization-whats-new-with-virtualization-from-red-hat

CentOS Convert 비교 요약

openmaru
오픈마루(주)

분류	CentOS To Red Hat Enterprise Linux	CentOS To OpenShift Container Platform
작업 절차	EOS 버전의 경우 2단계 이상 Upgrade 의 작업 절차	간소화
작업 시간	400시간 (VM 100대 기준)	7시간
자동화	자동화 없음 및 자동화 별도 구성 필요	자동화
복잡도	수동 작업으로 인한 복잡성 증대	자동화로 인하여 단순화
반복도	서버별 단위로 많은 수의 반복 필요	클러스터 단위로 적은 회수의 반복
OS	라이선스 구매 필요	포함
Runtime	라이선스 구매 필요	포함
Middleware	라이선스 구매 필요	포함
보안 소프트웨어	라이선스 구매 필요	포함

클라우드 네이티브 무상 컨설팅

찾아가는 클라우드 네이티브 세미나

세원아이티가 궁금해하는

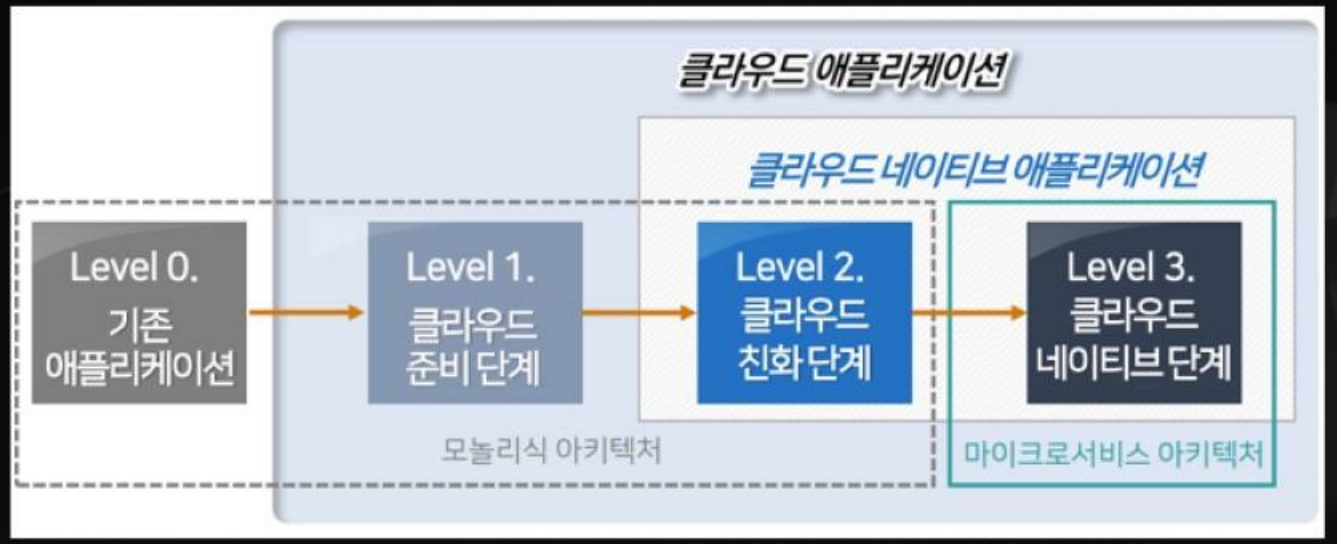
클라우드 네이티브 사전 질의 내용

공공기관의 클라우드 네이티브 적용 범위

클라우드 애플리케이션 성숙도 단계

openmaru
오픈마루(주)

- 클라우드 네이티브는 컨테이너 기반의 PaaS로 시작하여, CI/CD, MSA 모두 포함된 단계
- Lv1. 클라우드 준비 단계 → Lv2. 클라우드 친화 단계 → Lv3. 클라우드 네이티브 단계
  - PaaS와 컨테이너를 도입하는 클라우드 친화 단계
  - 데브옵스, CI/CD, MSA를 적용하는 클라우드 네이티브 단계



한국지능정보사회진흥원 클라우드 네이티브 발주자 안내서

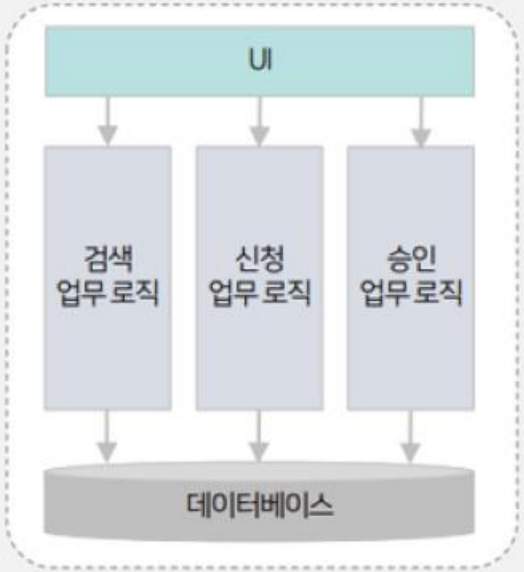
마이크로서비스 아키텍처로의 전환 예시

openmaru 오픈마루 (주)

한국지능정보사회진흥원 클라우드 네이티브 발주자 안내서

모놀리식 아키텍처

총무관리 시스템 - 회의실 관리 서비스



- 논리적 또는 물리적으로 웹[2], WAS[3], DB로 구성되어 수직적 인터페이스가 주로 발생

리팩터링 아키텍처

총무관리 시스템 - 회의실 관리 서비스

- 일부 서비스를 분리하여 별도 DB를 가지는 구조로 전환
- 서비스는 업무 로직 외에 서비스 연결을 위한 API가 필요함

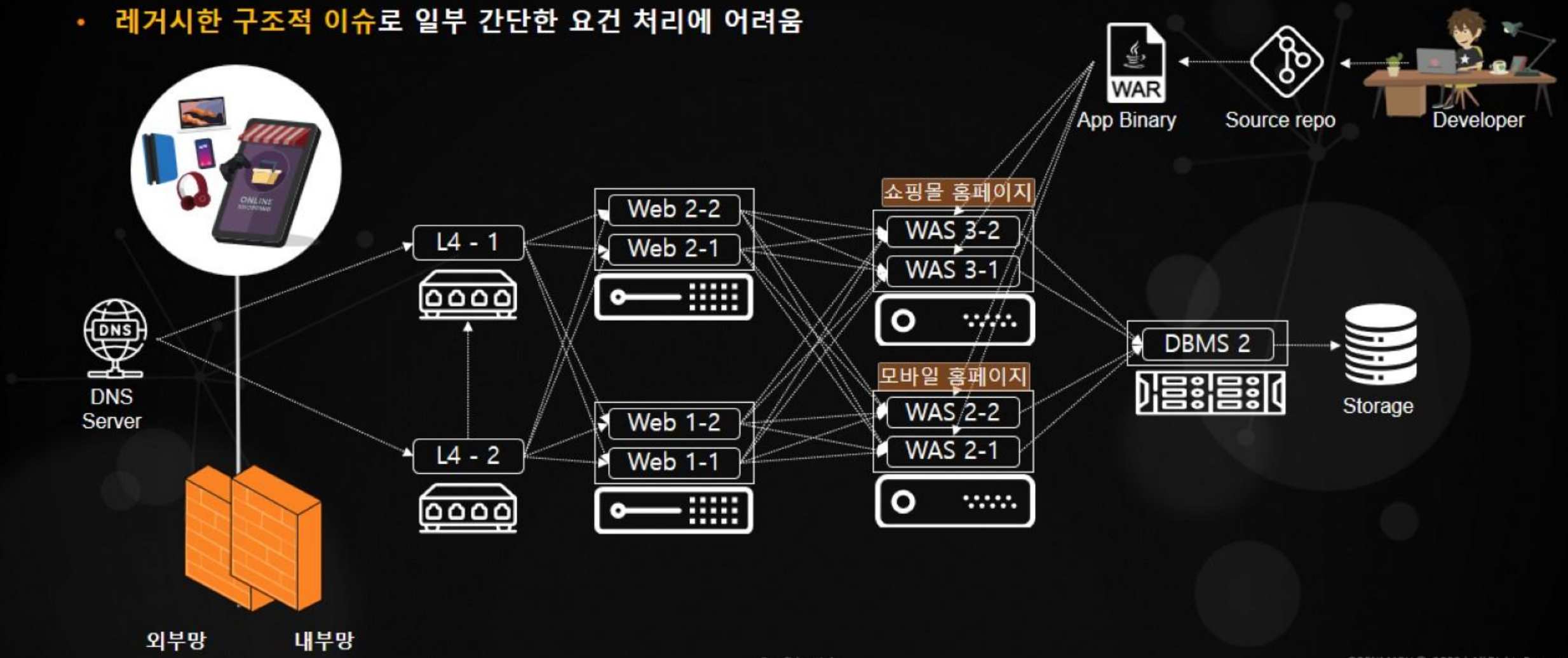
마이크로서비스 아키텍처

총무관리 시스템 - 회의실 관리 서비스

- 서비스별로 UI, API, 업무 로직, DB를 별도로 구성
- 해당 서비스별로 빌드-배포가 이뤄짐

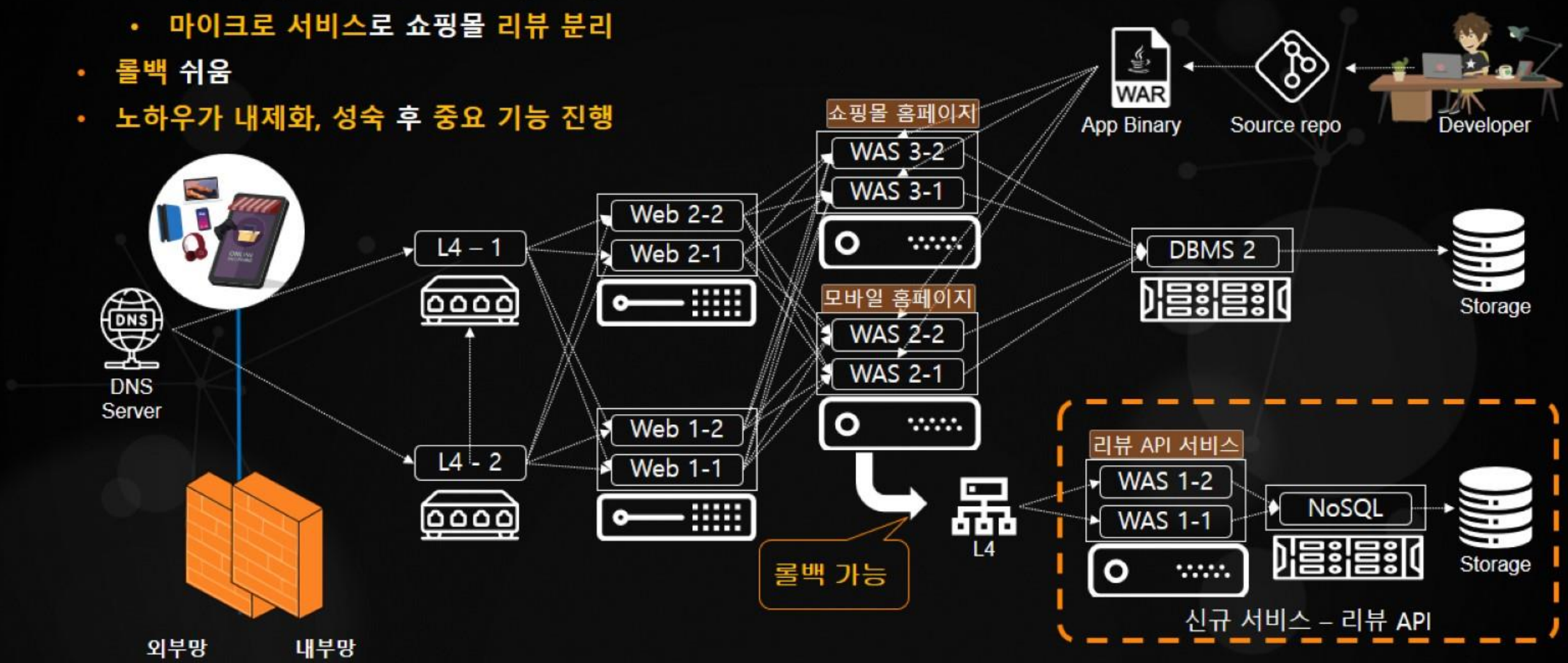
일반화 되어 있는 모놀리틱 아키텍처

- 웹서버, WAS 서버, 데이터베이스 의 역할 별로 티어를 나눈 3 티어 구조
  - 각 티어 별로 수동 확장과 관리, 오토스케일링 및 자동화 부족
- 레거시한 구조적 이슈로 일부 간단한 요건 처리에 어려움



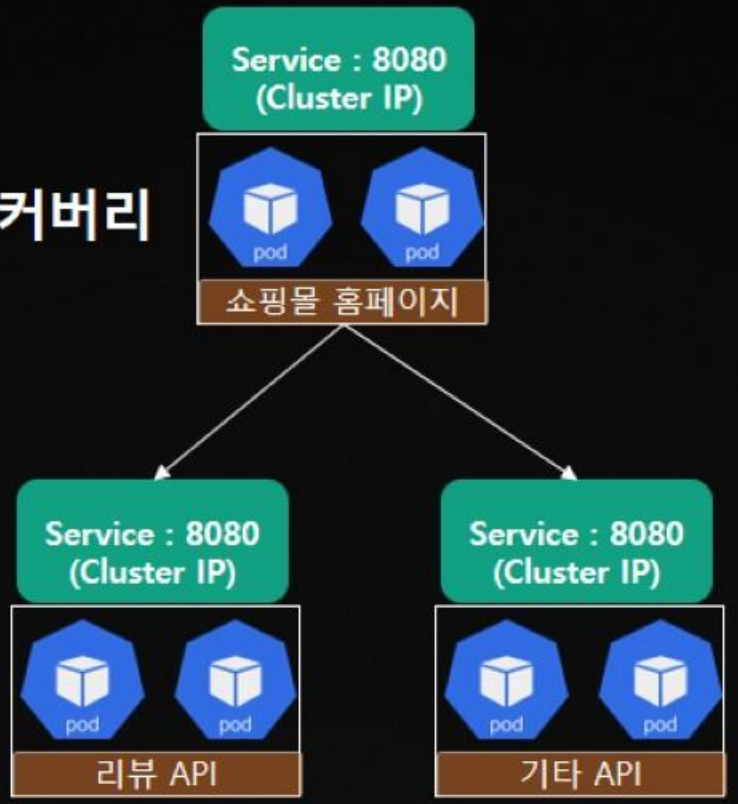
Microservice Pilot – 리뷰API 서비스 도입

- 수많은 기능으로 동작하는 기존 서비스에서 쉬운 부분부터 API로 분리
  - 재설계 시 리스크 부담으로 시도 어려움
  - 마이크로 서비스로 쇼핑몰 리뷰 분리
- 롤백 쉬움
- 노하우가 내제화, 성숙 후 중요 기능 진행



애플리케이션 배포 / 로드 발란서 / 서비스 디스커버리

- 서비스 (K8S, OpenShift)
  - 로드 발란서
  - Core DNS 기반 서비스 디스커버리

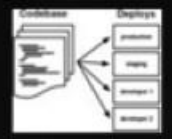


- Jenkins 필요 없음
- S2I(Source To Image) - GIT
  - Source 모드
    - Java
      - Maven, Gradle 등
      - WAS or Spring Boot 등
    - Nodejs, Python, PHP 등
  - Dockerfile 모드
  - Binary 모드
    - WAR, JAR , 드래그앤 드랍으로도 가능!!

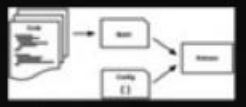
Cloud Native Application(애플리케이션 현대화) 도전!

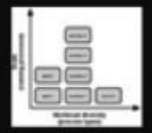
openmaru 오픈마루㈜

- 클라우드의 이점을 최대로 활용할 수 있도록 애플리케이션을 구축하고 실행하는 방식
- 신속한 개발과 클라우드 확장성 확보를 위한 클라우드 네이티브 애플리케이션 개발은 필수
  - SaaS 12-Factor, Cloud, MSA, Container, CI/CD 등 DevOps 중요
  - https://landscape.cncf.io/



12-Factor applications





Independent, autonomous, stateless, processes

Cloud Environment

NETFLIX

Configuration sources, service references

OpenJDK kafka

Microservices redis

CouchDB mongoDB.

Application as composition of polyglot services

CD.FOUNDATION Spinnaker

DevOps

JENKINSX

Build, package, deploy, observe

기존 애플리케이션과 Cloud Native 애플리케이션 비교

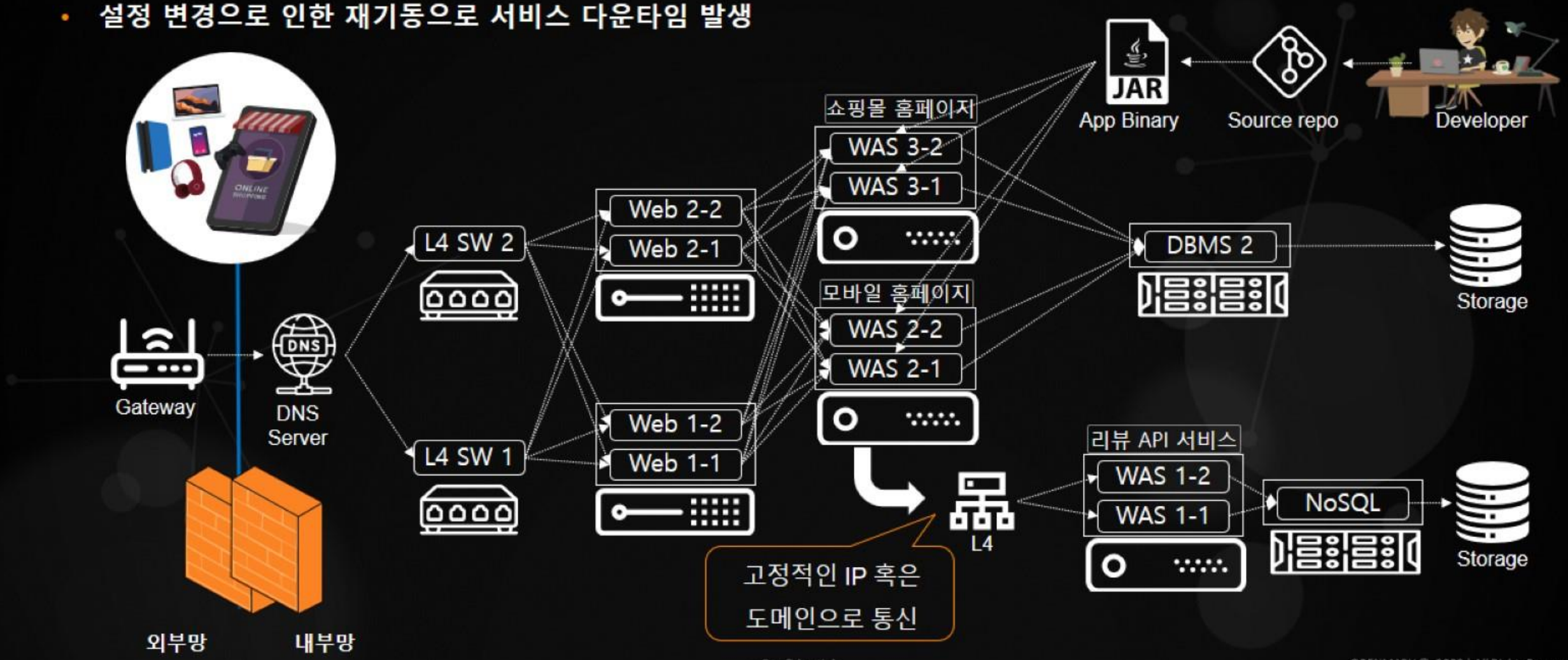
openmaru
오픈마루(주)

분류	기존 애플리케이션	Cloud Native 애플리케이션
실행 환경	물리 서버 중심	컨테이너 중심
확장	Scale Up (수직 확장)	Scale Out (수평 확장)
결합	크고 조밀 결합	느슨하게 & 서비스 기반
인프라 의존성	인프라 의존	인프라 독립적으로 이동성 보장
Delivery 방법	폭포수형으로 장기간 개발	Agile & Continuous Delivery
개발 도구	로컬 IDE 개발 도구	클라우드 기반의 지능형 개발 도구
조직구조	사일로화 된 개발, 운영, 보안 팀	DevSecOps, NoOps 또는 협업

클라우드 네이티브 전환 시 고려사항

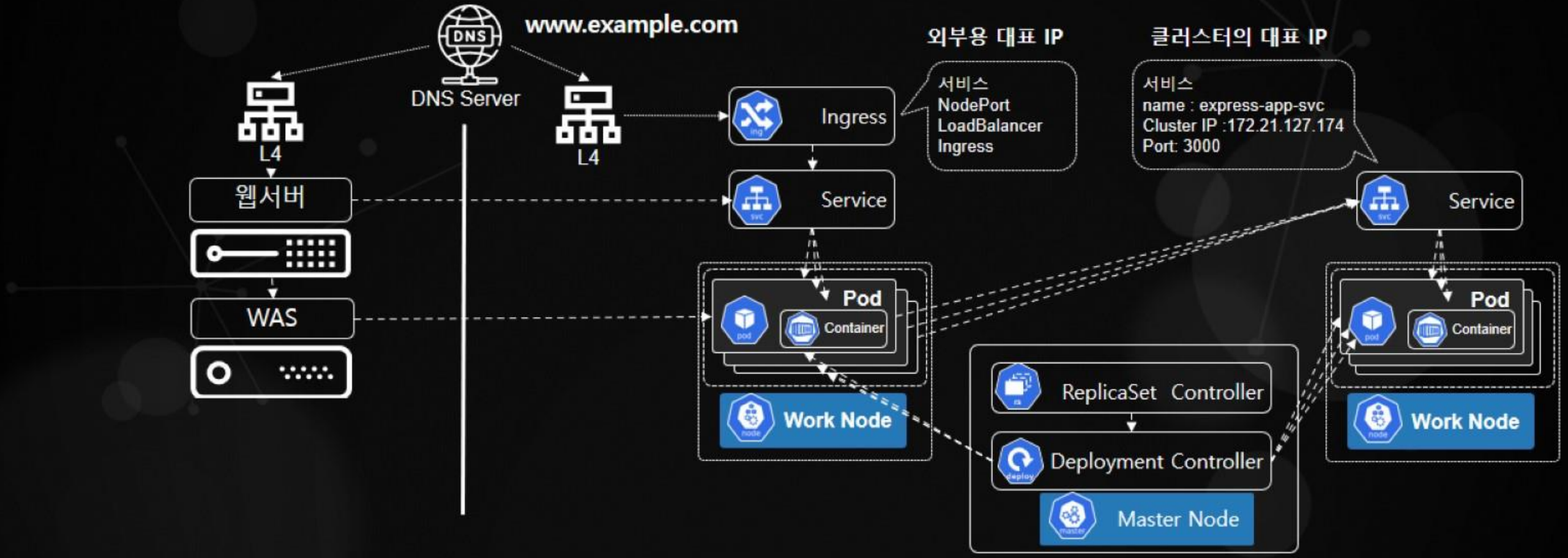
AS-IS(레거시) 환경의 고정IP를 이용한 애플리케이션간 통신

- 서비스 확장 시 모든 설정을 수작업
- 설정 변경으로 인한 재기동으로 서비스 다운타임 발생



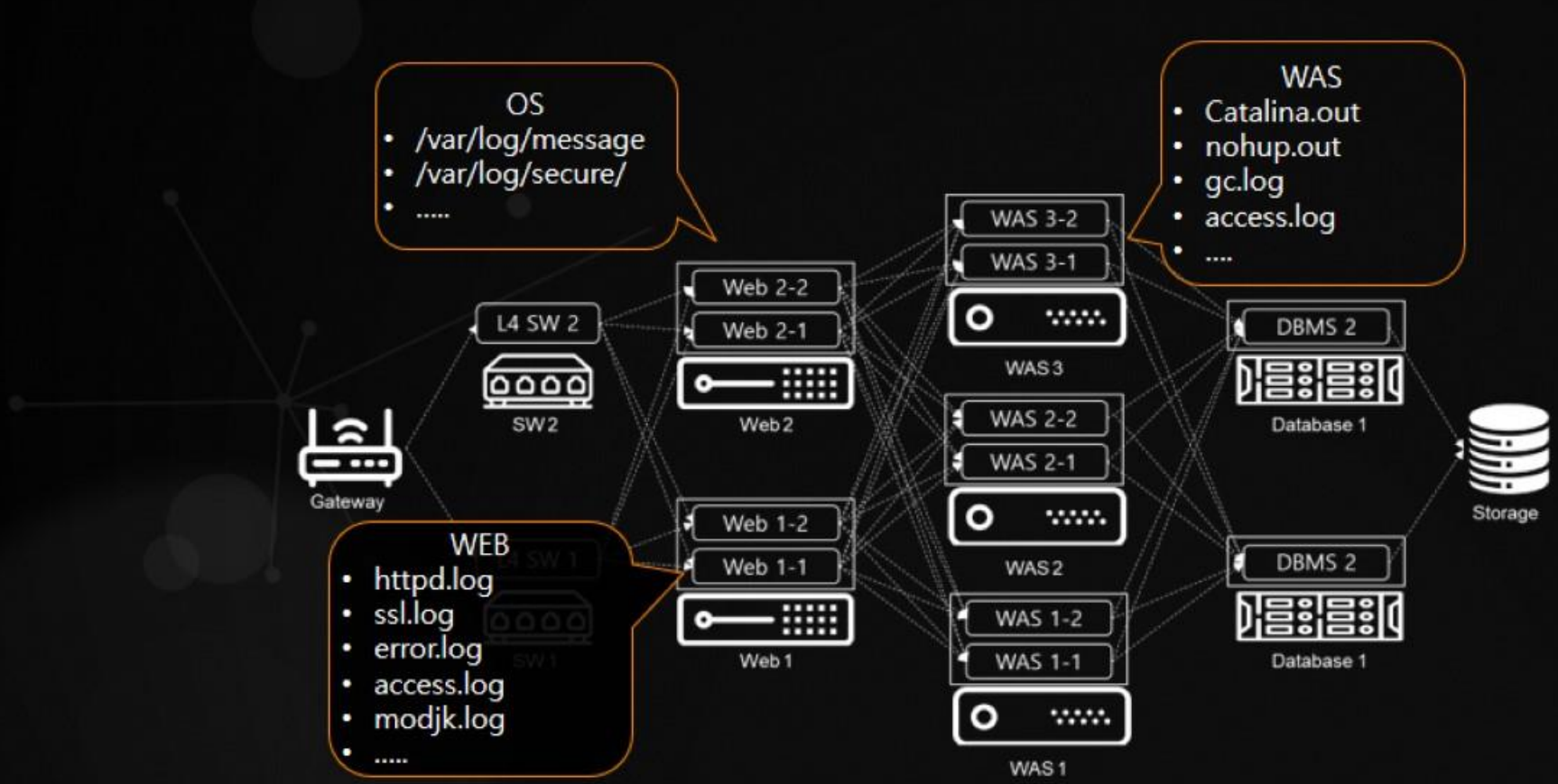
컨테이너 간의 호출은 service를 이용

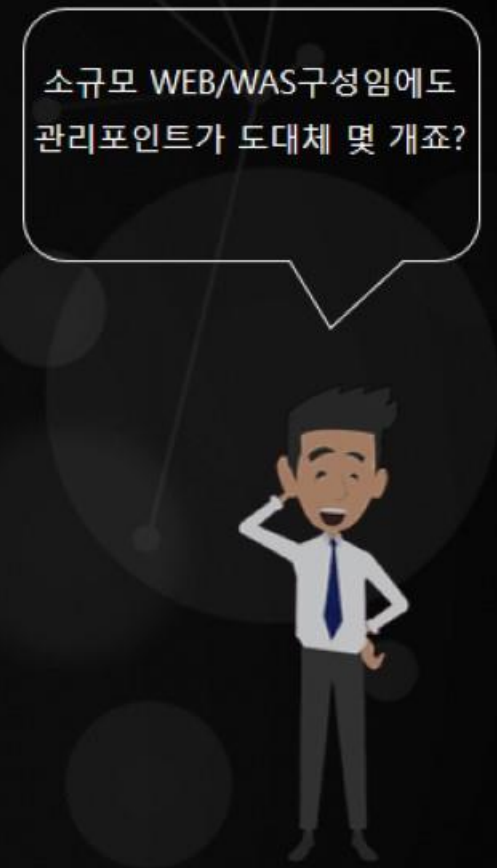
- 컨테이너 환경의 내부 Load Balancer(Domain 기반) 통신
- Service Object의 도메인 기반 으로 호출
- Pod는 유동적인 IP Pool을 갖으며 고정적인 IP를 가지지 않음



AS-IS – 개별 서버 별로 로그 파일 관리

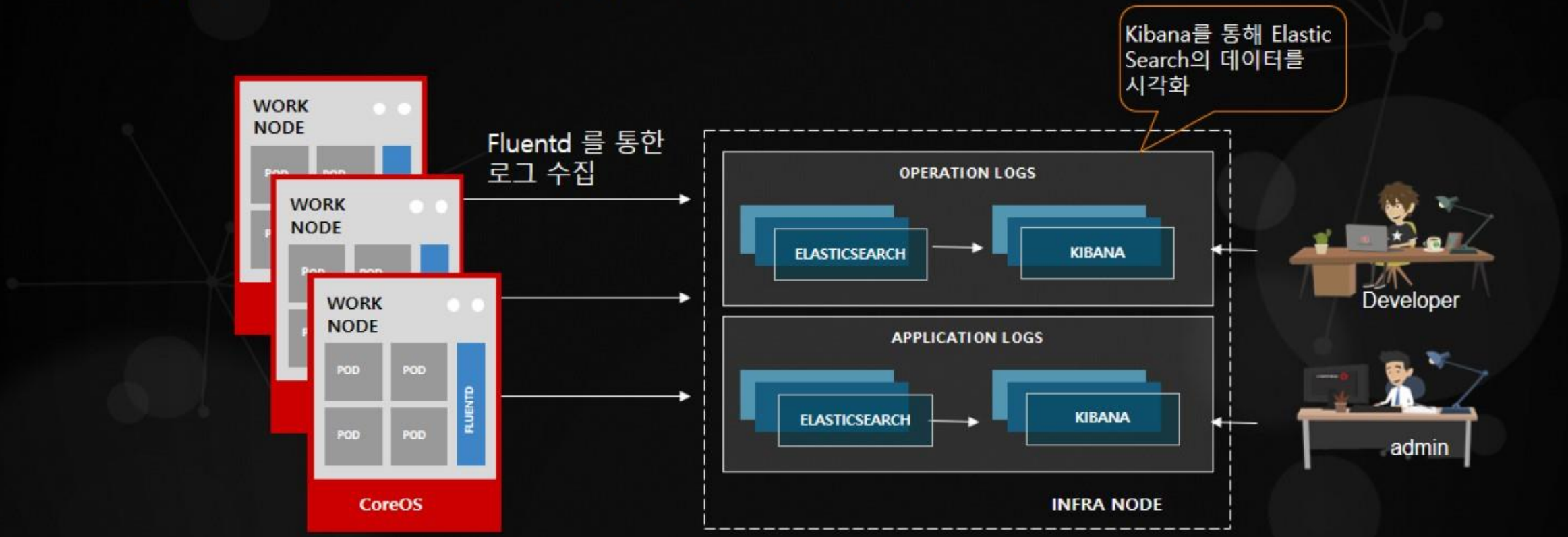
- 발생하는 로그들이 각각의 서버 Local Disk에 저장됨
- 에러 발생시 에러가 발생한 서버를 찾아야 하고 발생한 에러로그를 분석
- 서버 대수가 늘어날 수록 모니터링 해야하는 Log의 수도 늘어남





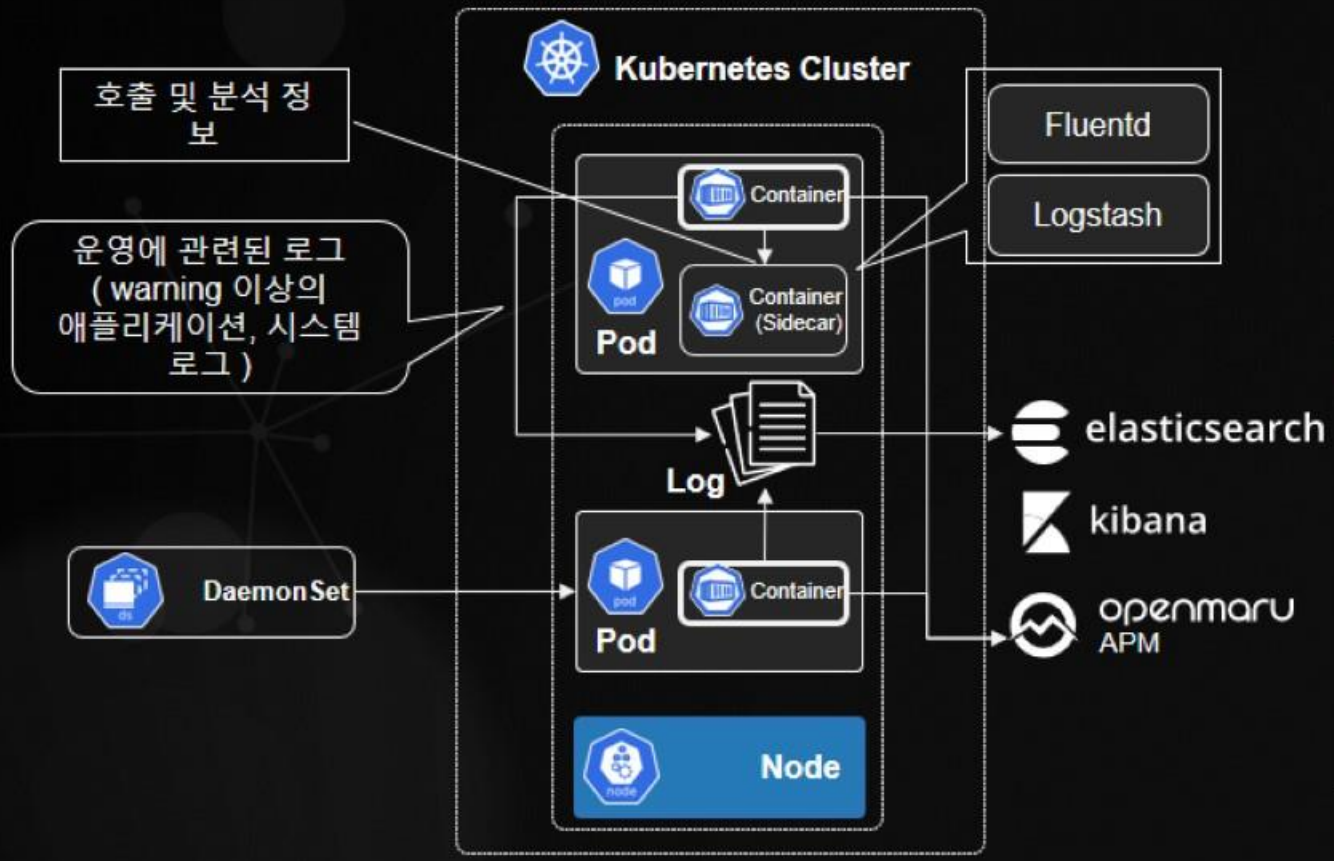
To-Be : 통합 로그 관리

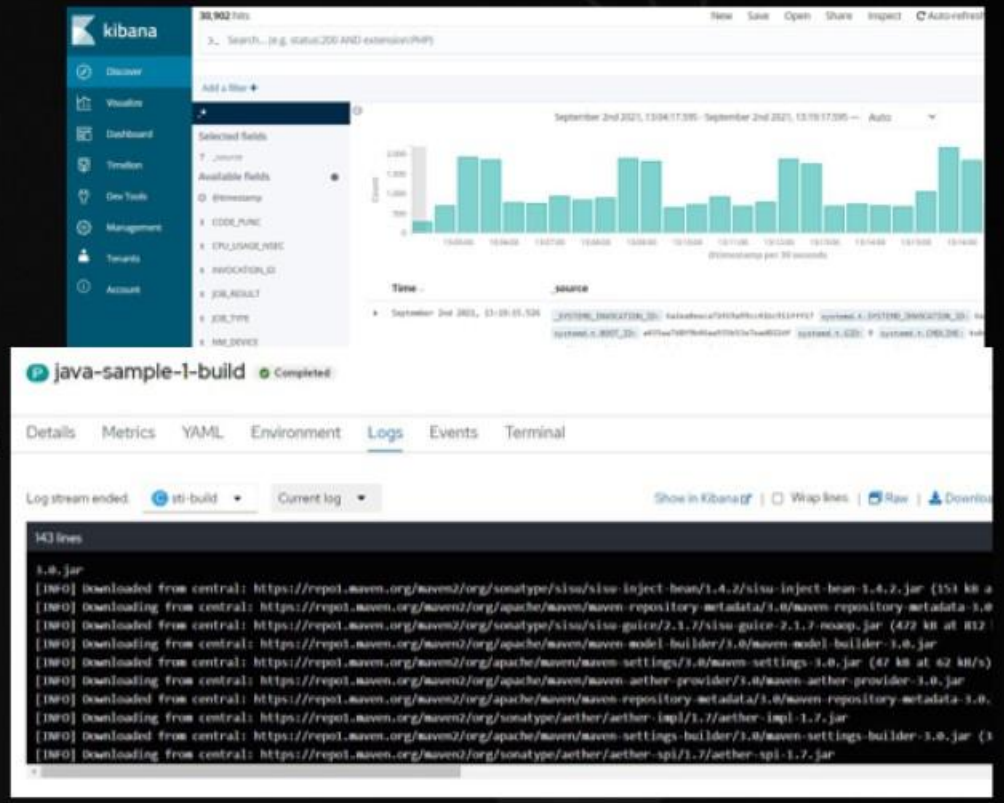
- PaaS 플랫폼의 경우 약 200개 이상의 플랫폼 인프라 컨테이너 기동
- 애플리케이션 컨테이너도 기존 환경에 비해 몇배 많은 수가 기동
- 수많은 컨테이너의 로그를 통합 모니터링하는 스택이 필수
- EF(L)K (Elastic Search + Fluentd(Logstash) + Kibana) 스택



To-Be : 통합 애플리케이션 로그 관리 방안

- 기존 레거시 환경에서는 개별 서버에 로그 생성 후 별도 작업으로 통합 관리
- EFK (ElasticSearch, Fluentd, Kibana) 를 활용한 자동 로그 통합 관리
  - 애플리케이션의 로그 정책을 STDOUT으로 설정





클라우드 네이티브 환경의 애플리케이션 모니터링 어려움

openmaru
오픈마루㈜

6. Monitor, log and troubleshoot from the start

The construction of applications from a set of microservice Legos considerably complicates how to monitor and troubleshoot systems and their performance. Various microservices often trigger a cascade of events that leads to an application failure. To minimize failures -- which aren't a *maybe*, but a reality -- incorporate monitoring and troubleshooting into microservices design.

https://www.techtarget.com/searchitoperations/tip/Follow-these-6-steps-to-deploy-microservices-in-production

- 마이크로 서비스 아키텍처일수록 **모니터링 방법이 상당히 복잡**해 집니다.
- 마이크로 서비스 아키텍처에 **모니터링과 트러블 슈팅 방법이 고려**되어야 합니다.

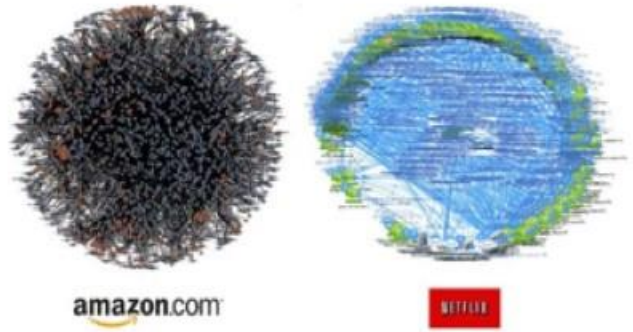
- Uber는 2014년 말 에 4,000개가 넘는 독점 마이크로 서비스와 점점 더 많은 수의 오픈 소스 시스템이 모니터링 시스템에 문제를 제기했다고 보고
- 컨테이너 인프라는 모니터링 시스템이 필수적인 환경
- **마이크로 서비스에 특화된 모니터링 시스템이 필요**

2. Microservices instead of a monolith

Following a microservice architecture, a typical monolith application would be broken down into a dozen or more microservices, each one potentially running its own programming language and database, each one independently deployed, scaled and upgraded.

Uber for example reported in late 2014 over 4,000 proprietary microservices and a growing number of open source systems which posed a challenge for their monitoring system.

The Challenge: A surge in the number of discrete components you need to monitor.

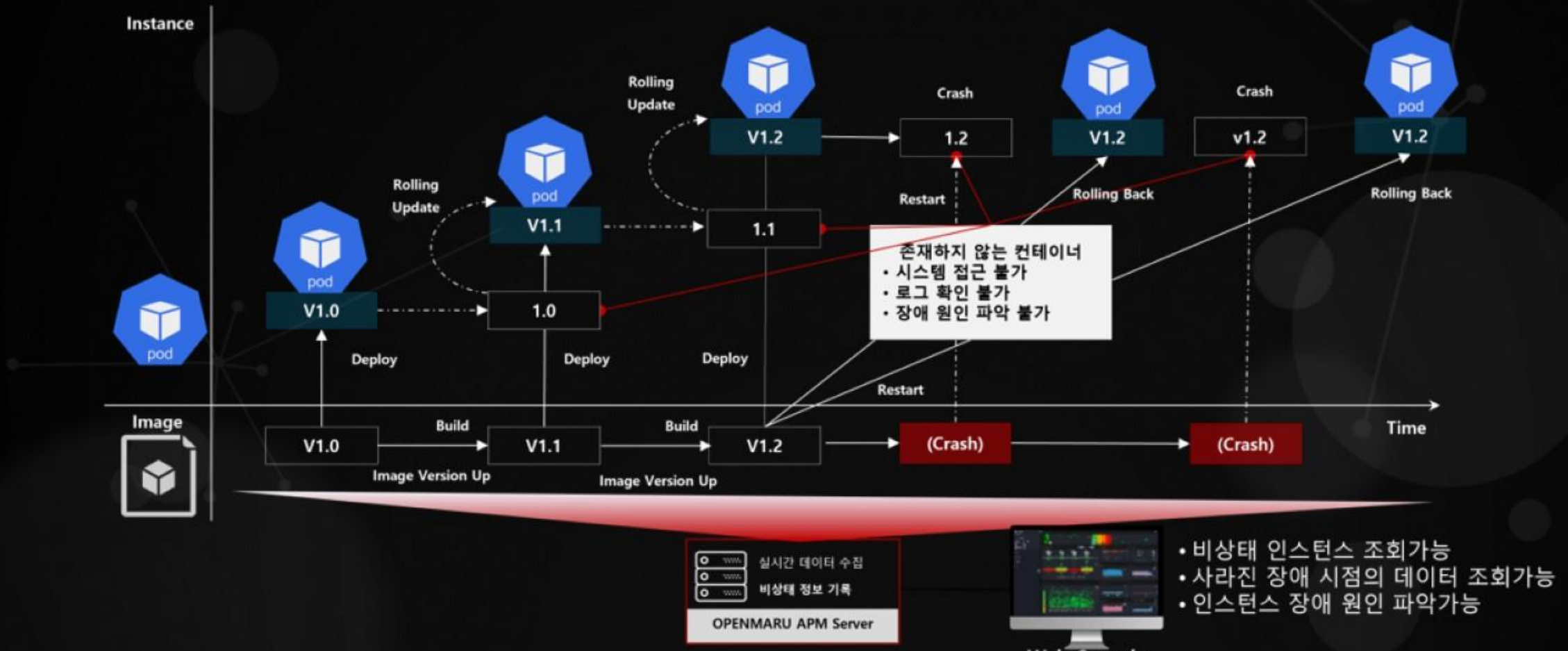


https://www.infoq.com/articles/microservice-monitoring-right-way

컨테이너는 무상태 (Immutable) 인프라스트럭처로 상태를 저장하지 않음

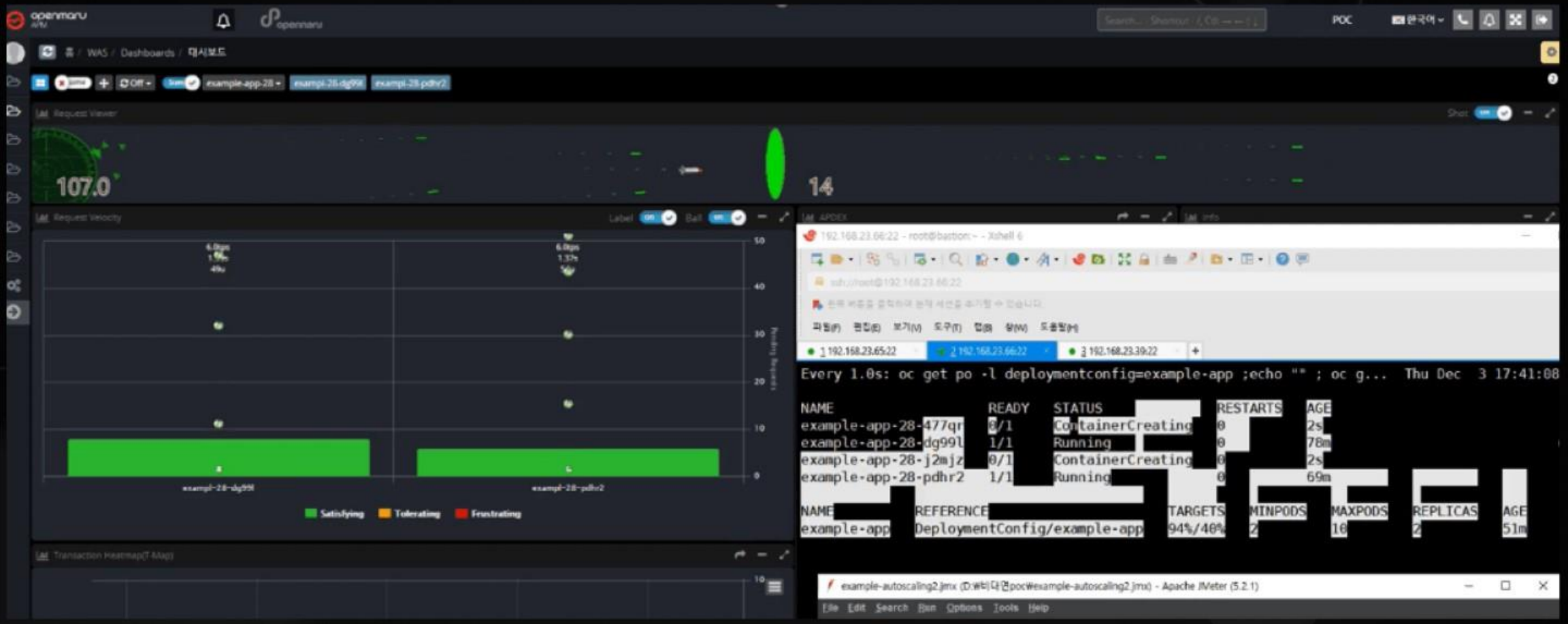
openmaru 오픈마루(주)

- PaaS 환경에서 컨테이너가 중지된 후 장애원인 파악을 위한 방법을 제공
- APM 에서 사라진 컨테이너에 대한 정보를 보관하여 장애원인을 파악할 수 있음



컨테이너 환경에서의 애플리케이션 모니터링

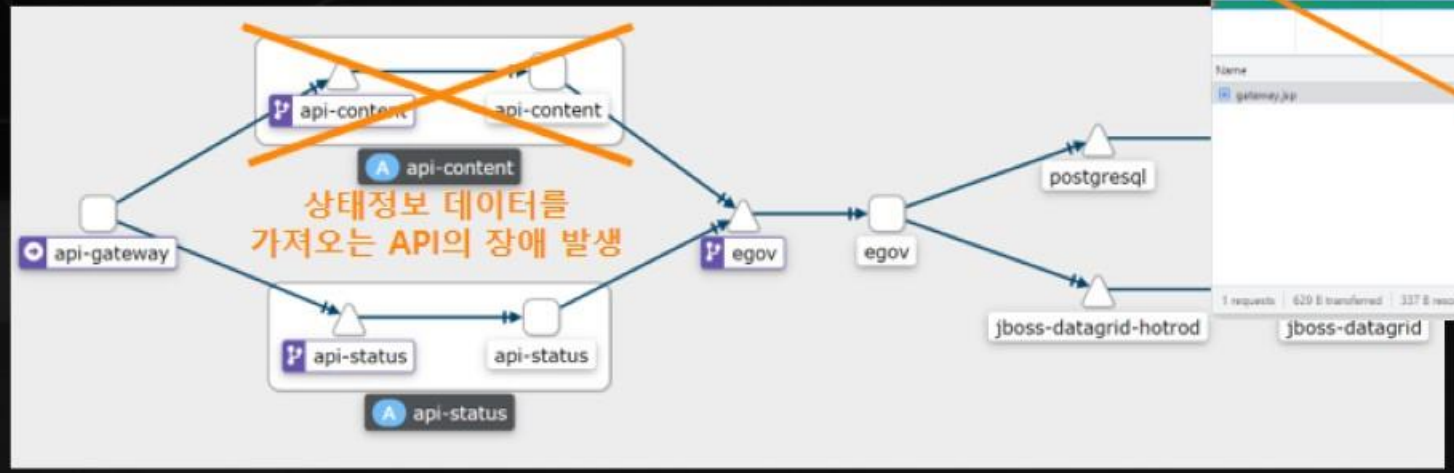
- **Auto Scaling으로 Container(Pod)가 유연하게 Scale Out/In이 발생하는 환경**
- **클라우드 네이티브 애플리케이션을 트러블 슈팅할 수 있는 기능이 있는 모니터링 시스템이 필요함**

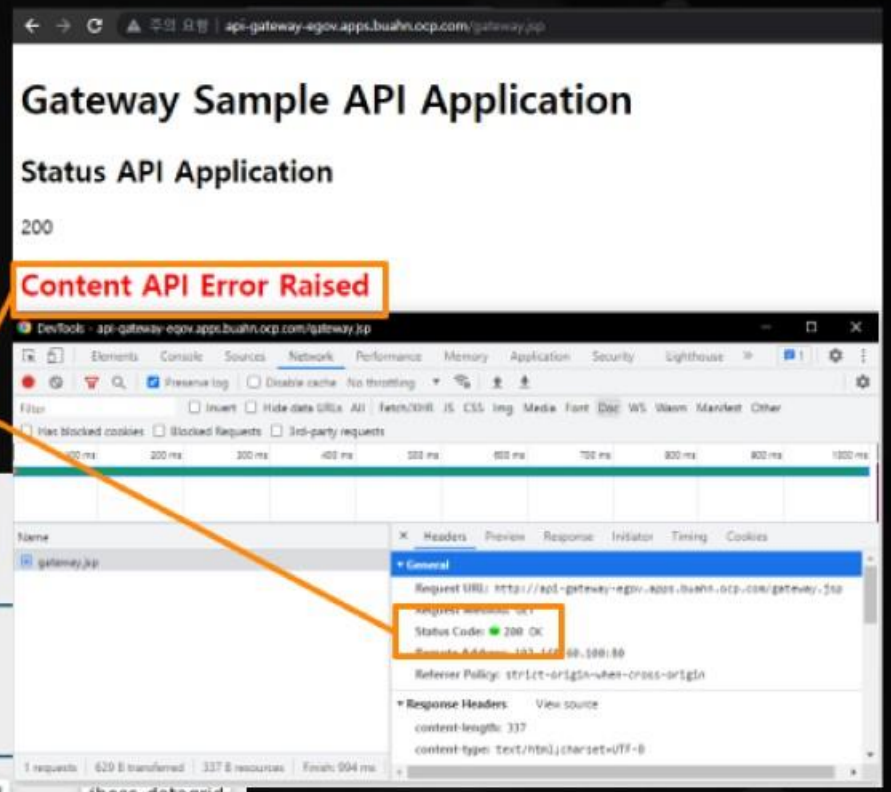


일부 애플리케이션 장애 발생

- MSA 환경의 특성상(느슨한 결합) 일부 서비스의 장애가 발생했을 때 식별이 어려움
  1. gateway를 접속하더라도 정상 페이지(status 200) 출력
  2. Content API로 받아오는 데이터는 에러 발생
- MSA에서 장애가 발생한 애플리케이션을 식별 및 분석에 어려움

Content API에서는 장애가 발생했지만
Gateway 애플리케이션은 200 상태코드 반환

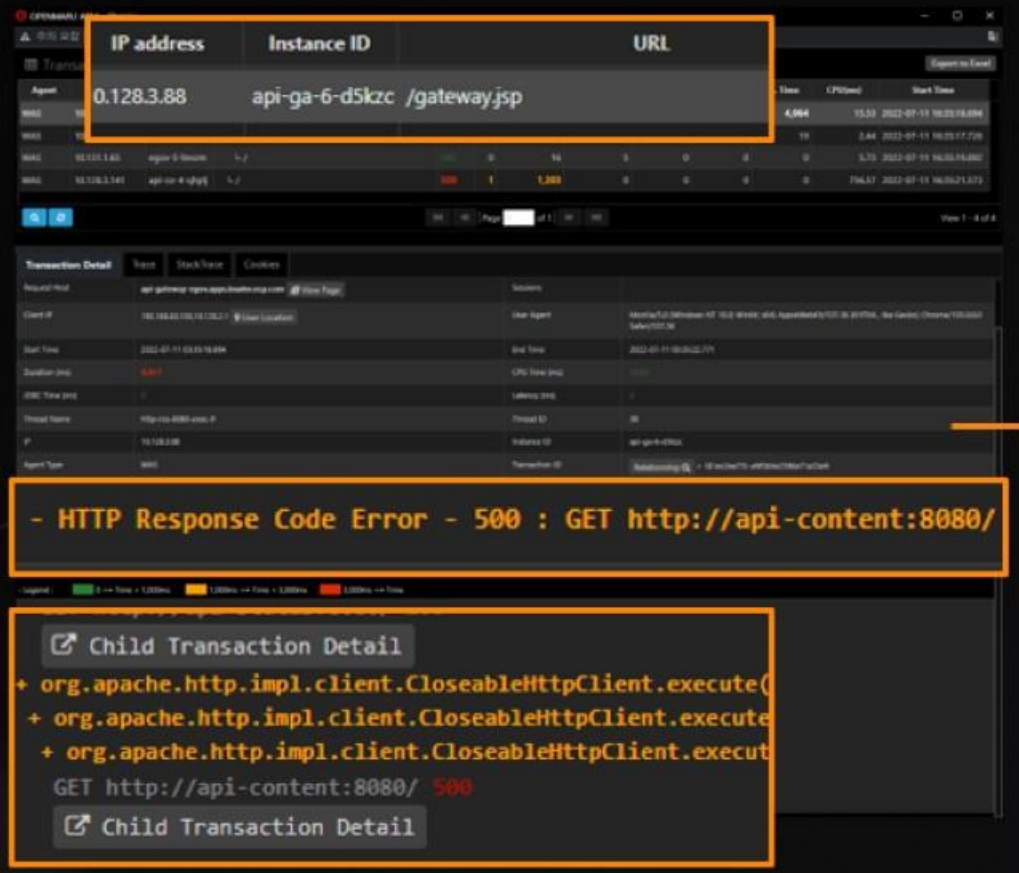


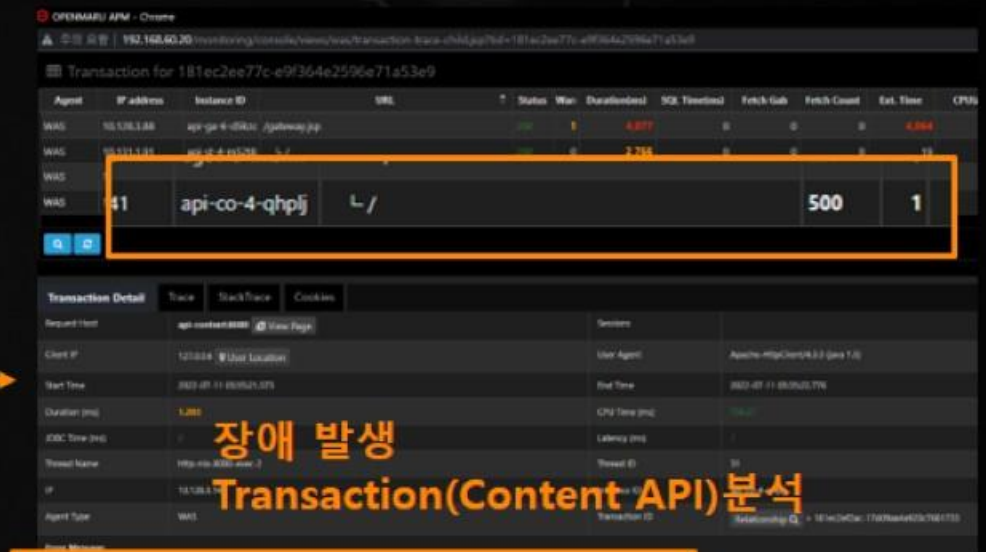


장애 발생 애플리케이션의 분석 절차

openmaru
오픈마루㈜

장애 발생
트랜잭션을 드래그





장애 발생
Transaction(Content API)분석

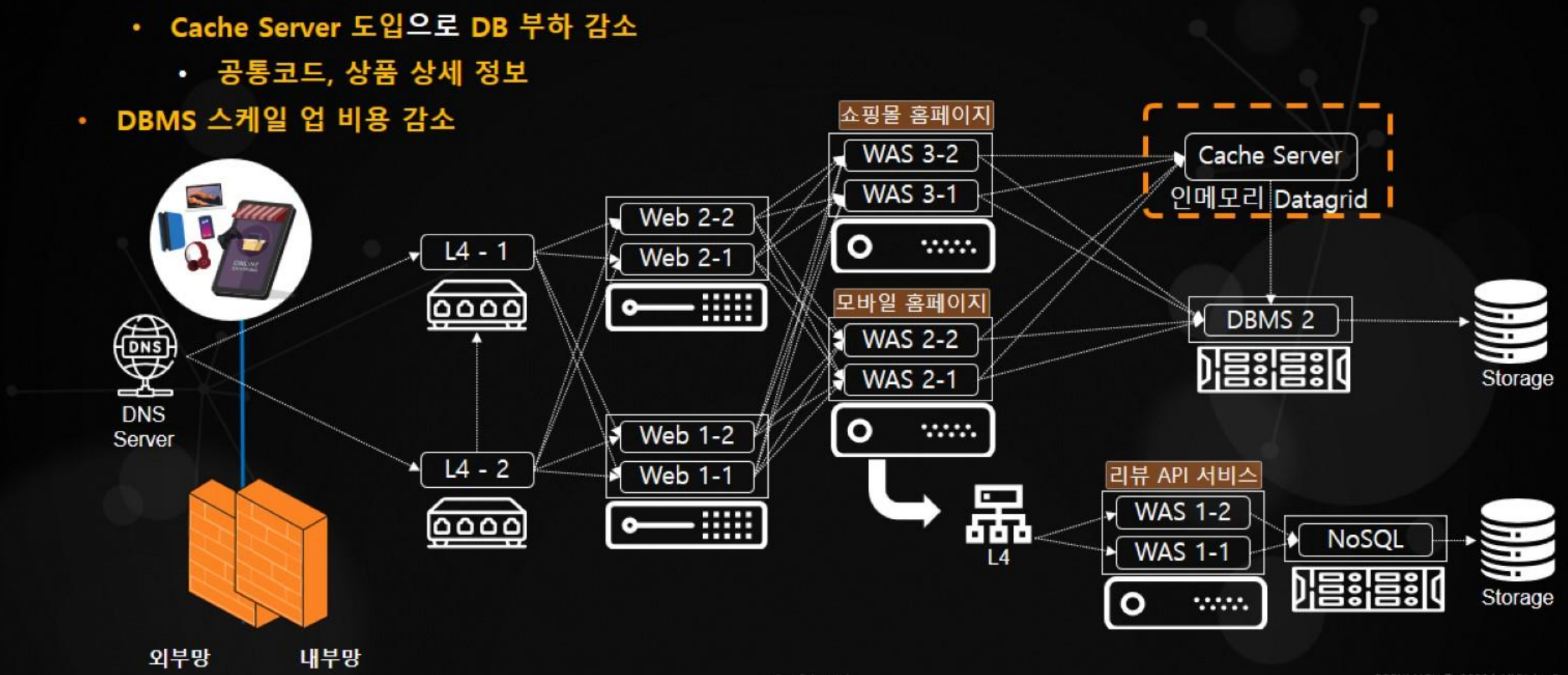
```
    Unable to compile class for JSP:

An error occurred at line: [21] in the jsp file: [/index.jsp]
testtesttest cannot be resolved to a type
18:     } catch ( Exception e ) {
19:         e.printStackTrace();
20:     }
21:     testtesttest
22:     %>
23:     < h2>Content API Application< /h2>
24:     < %
```

대규모 사용자을 위한 성능 대응 방안 – 엔터프라이즈 캐쉬 서버 도입

openmaru 오픈마루㈜

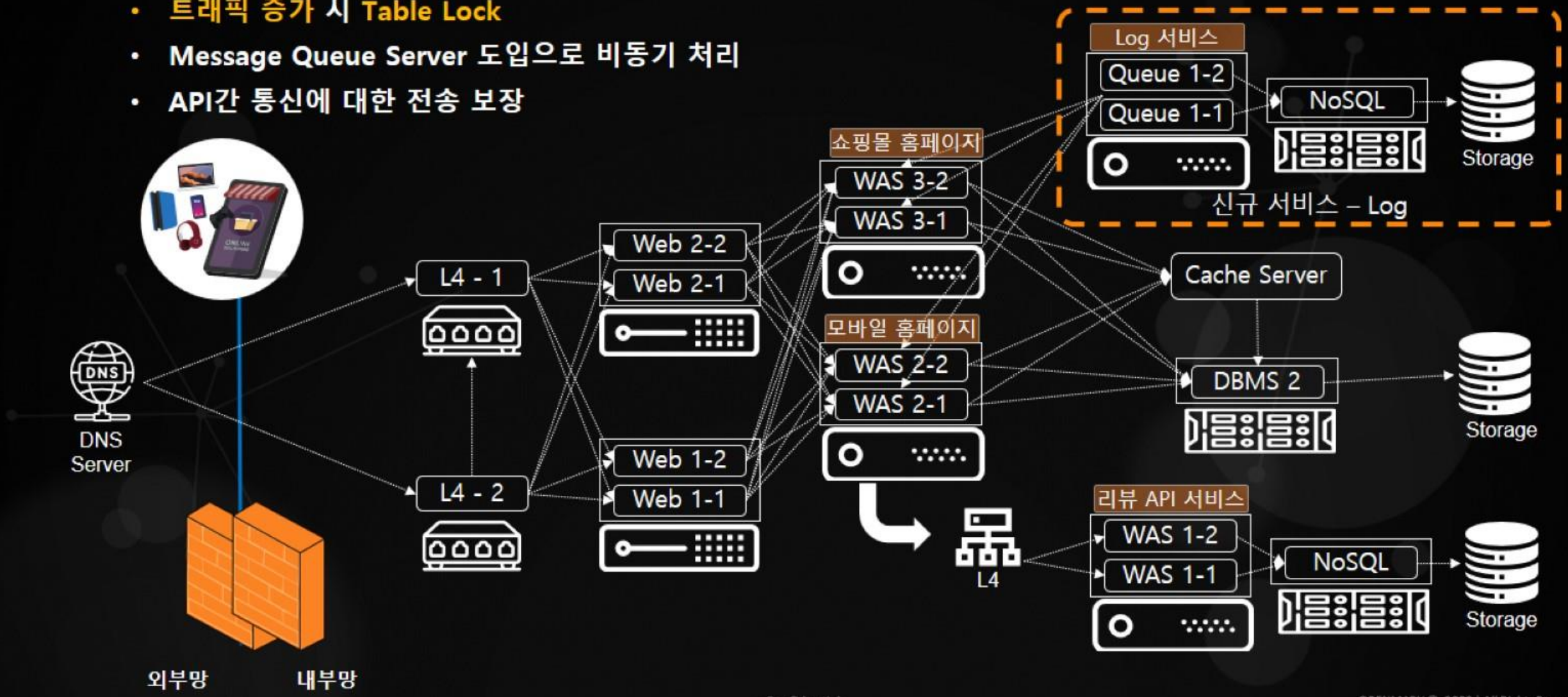
- 웹서비스에서 DB 요청 중 읽기 비율은 약 80% 이상
- 서비스 사용자 증가로 인한 성능 개선과 고가용성 확보
  - Cache Server 도입으로 DB 부하 감소
    - 공통코드, 상품 상세 정보
- DBMS 스케일 업 비용 감소



대규모 사용자을 위한 성능 대응 방안 – 비동기 처리 메시지 큐

openmaru 오픈마루㈜

- 보안에 중요한 로그인 히스토리 및 중요 로그 DB화
  - 트래픽 증가 시 Table Lock
  - Message Queue Server 도입으로 비동기 처리
  - API간 통신에 대한 전송 보장

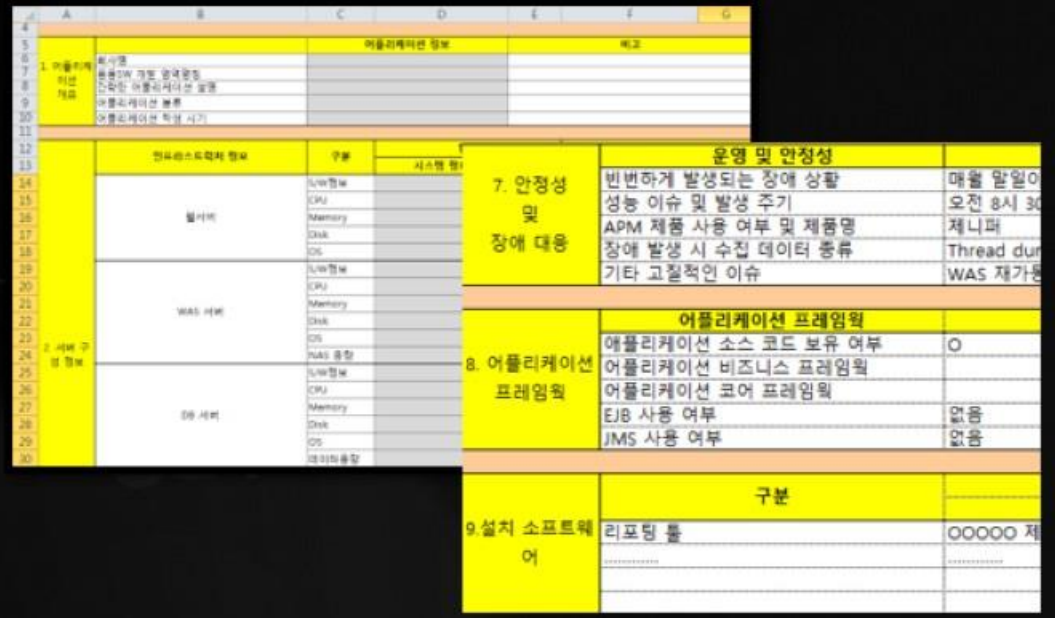


클라우드 네이티브 적용 가능 업무

CentOS To OpenShift 난이도 분석 방안

애플리케이션의 마이그레이션이 가장 중요 !

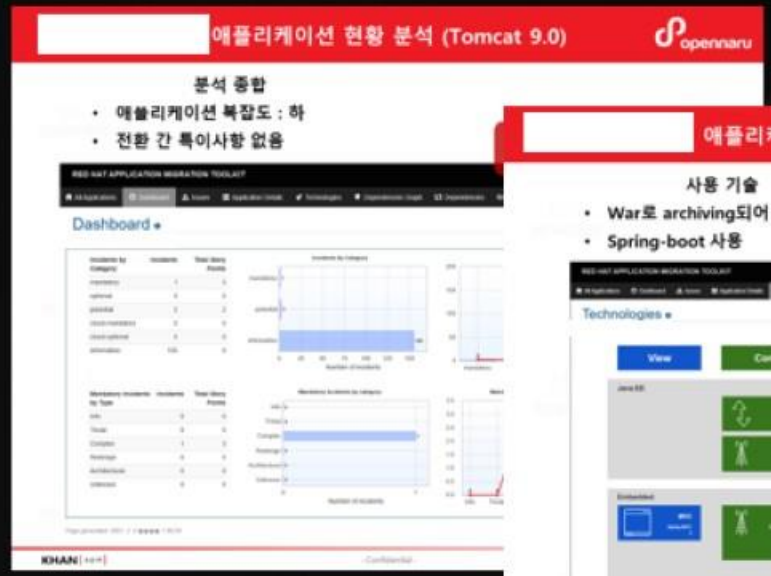
- 애플리케이션의 컨테이너화가 가능 유무, 난이도, 필요 공수 등을 분석이 반드시 필요.
- 일부 시스템에 종속되는 애플리케이션의 경우 마이그레이션이 불가능할 수 있음.
- 별도의 진단 도구를 활용한 분석 수행



7. 안정성 및 장애 대응	운영 및 안정성	
	빈번하게 발생되는 장애 상황	매월 말일
	성능 이슈 및 발생 주기	오전 8시 3
	APM 제품 사용 여부 및 제품명	제니퍼
	장애 발생 시 수집 데이터 종류	Thread du
	기타 고질적인 이슈	WAS 재가

8. 어플리케이션 프레임워크	어플리케이션 프레임워크	
	애플리케이션 소스 코드 보유 여부	O
	어플리케이션 비즈니스 프레임워크	
	어플리케이션 코어 프레임워크	
	EJB 사용 여부	없음
	JMS 사용 여부	없음

9.설치 소프트웨어	구분	
	리프팅 툴	OOOOO 제
	.....	.....



Cloud Native 애플리케이션에 적합한 업무 조건

openmaru
오픈마루(주)

Cloud Native 애플리케이션에 적합한 조건

- JAVA 기반의 애플리케이션
- UNIX/Windows에 종속되는 라이브러리를 사용하지 않음
- 외부 연계 최소화
- 상용 솔루션 최소화
- 결제 모듈과 같이 복잡한 내부 처리 로직 최소화
- 대민 서비스로 트래픽 부하가 유동적
- 데이터 사이즈가 크지 않은 애플리케이션
- 오픈소스 WEB/WAS를 사용하는 애플리케이션
- 소스코드를 보유하고 있는 애플리케이션
- 상용 솔루션 벤더의 지원을 받을 수 있는 애플리케이션

Cloud Native 애플리케이 Check List

openmaru
오픈마루(주)

Check Point	대상 유무	비고
애플리케이션이 JAVA 기반으로 작성되었는가?		C와 같이 머신에 종속되는 언어일 경우 마이그레이션에 어려움. 최신 버전의 JAVA 버전(1.8 이상)사용할 경우 적합
UNIX 혹은 Windows에 종속되는 라이브러리 및 솔루션을 사용하고 있지 않은가?		IIS와 같이 특정 OS에 종속된 소프트웨어를 사용하는 시스템은 마이그레이션에 어려움
외부 연계가 최소화 되거나 별도의 외부 연계 솔루션을 사용하는 애플리케이션인가?		외부 연계를 솔루션을 이용하면 난이도 하락
보안 소프트웨어와 같은 상용 솔루션 등이 최소화된 애플리케이션인가?		보안 소프트웨어와 같은 상용 솔루션들이 컨테이너화 되는지 벤더에 반드시 확인 필요
결제 모듈과 같이 복잡한 내부 처리 로직이 포함되지 않은 애플리케이션인가?		내부 처리 로직 및 내부 연계가 많을 경우 마이그레이션에 어려움
대민 서비스를 위한 애플리케이션으로 트래픽이 유동적인가?		트래픽이 유동적이거나 대민 서비스일 경우 Cloud Native에 적합한 애플리케이션
애플리케이션이 사용하는 데이터 사이즈가 크지 않은 애플리케이션인가?		빅데이터 플랫폼과 같이 사용하는 데이터가 큰 애플리케이션의 경우 마이그레이션 어려움
Apache HTTPD/Apache Tomcat과 같은 오픈소스 WEB/WAS를 사용하는 애플리케이션인가?		오픈소스 WEB/WAS를 사용하는 애플리케이션일 경우
Red Hat Linux와 같은 Linux에서 동작중인 애플리케이션인가?		CentOS의 EOS에 대응하여 엔터프라이즈 Linux로 변경이 필요한 경우
소스코드를 보유하고 있어서 기관에서 관리가 가능한 애플리케이션인가?		온나라 시스템과 같은 업무는 중앙부처에서 관리하는 애플리케이션으로 마이그레이션 불가
상용 솔루션에 대한 벤더 업체에 지원을 받을 수 있는 애플리케이션인가?		상용 솔루션의 경우 벤더 업체 이외에는 마이그레이션 불가

Cloud Native 애플리케이 Check List

openmaru
오픈마루(주)

Cloud Native에 적합한 애플리케이션 예시

- 대민 포탈 업무
- 기관 대표 홈페이지
- 그 외 홈페이지성 업무
- 전자정부프레임워크 기반 애플리케이션

기타 참고 사항

- 조건을 충족하지 않는 애플리케이션들은 별도의 진단 및 담당자와의 인터뷰를 통한 적합성 분석 필요.
- 조건을 충족하는 JAVA 기반의 애플리케이션이더라도 진단을 통해 난이도가 높게 나타날 수 있음.
    - EJB, WebServices, 상용 솔루션 갯수 등의 영향

클라우드 네이티브 보안 문제 해결 방안

레거시 환경에서 필요한 서버 보안

- OS 보안, 서버 접근제어 등 보안 5종 S/W를 OS 설치
  - OS 마다 설치하기 때문에 물리서버는 1Copy이고, 가상화는 VM 개수 만큼 설치
- 서비스에 따라 부가적인 보안 소프트웨어 필요(개인정보 검출, 비정형 암호화 등)
- 법적인 근거로 인해 서버보안 – 전자금융법, ISMS 등의 요건

컨테이너를 기동시키기 위한 CoreOS

Immutable Infra Structure, Red Hat CoreOS

기존 OS에 필요한 Host OS 보안 프로그램

- 취약점 스캐너
- 서버접근제어
- OS 보안
- 백업 Agent
- 로그 수집
- VM 혹은 Baremetal 대수 만큼 필요

Red Hat CoreOS는 Immutable Infrastructure

- RHCOS는 플랫폼에 포함된 Compliance 시스템
  - 1금융권 사례
- Immutable Infrastructure 특성으로 Read-Only OS

컨테이너 환경에서 필요한 보안

openmaru
오픈마루(주)

왜 AS-IS 환경의 보안들이 필요하지 않을까?

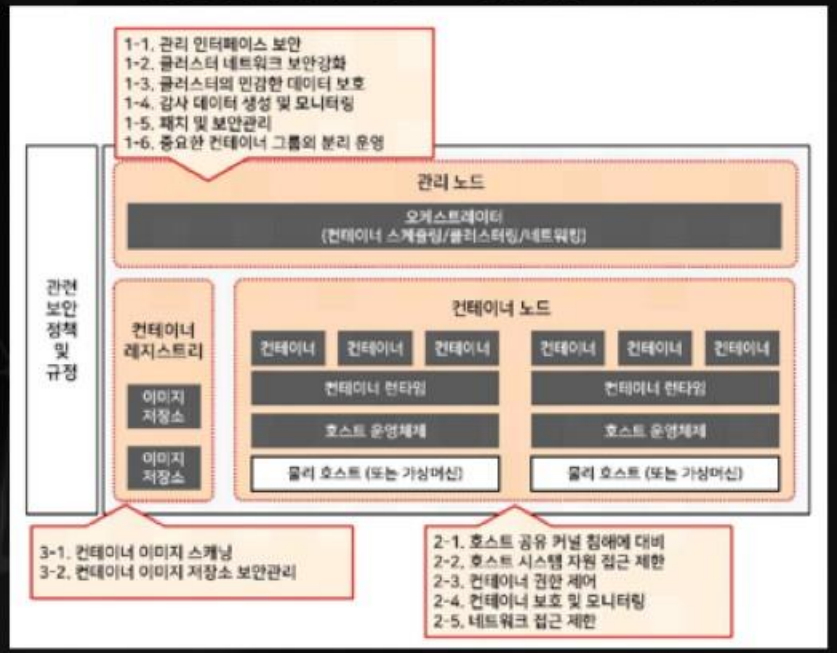
실질적으로 컨테이너환경에서 필요한 보안들

- 컨테이너 실행 권한에 대한 보안 : SCC
  - Container breakout
- 취약한 Container Image 사용
  - 악성코드, 채굴 프로그램이 포함된 이미지
- Role Base Access Controle : RBAC
  - 사용자별 필요 권한 부여
- 컨테이너 플랫폼의 Audit 로깅
  - EFK
- 신뢰할 수 있는 컨테이너 런타임 보안
  - Capabilities, SELinux, Seccomp & Namespace
- 컨테이너간의 네트워크 격리
  - Network Policy

국정원의 컨테이너 기술 보안 기준

- 컨테이너 기반 정부 업무 인프라의 안정성과 신뢰성을 제고하기 위한 보안기준
- 보안 기준은 관리노드, 컨테이너 노드 및 컨테이너 레지스트리로 분류
- 각급 기관은 컨테이너 기반의 업무 서비스 도입 및 운영 시 해당 보안 기준 내용 준수

국정원 컨테이너 기술 스택에 대한 보안 기준



국정원 컨테이너 기술 점검항목

5. 컨테이너 가상화 기술 보안기준 및 점검항목

○ 관리 노드

점검항목	중요도	점검결과
1-1. 관리 인터페이스에 대한 인증과 접근 제어를 강화해야 한다.		
(1) 관리자 계정 접근에 대해 다중요소(Multi-Factor) 인증을 사용하는가?	필수	
(2) 네트워크를 통한 관리 인터페이스 접속 시에는 암호화 프로토콜을 사용하고 있는가?	필수	
(3) RBAC(역할 기반 접근 제어, Role Based Access Control) 정책을 적용하고 있는가?	권고	
1-2. 클러스터의 네트워크 보안을 강화해야 한다.		
(1) 클러스터 관리 구성요소에 대한 불필요한 네트워크 접근을 차단하고 있는가?	필수	
(2) 클러스터 네트워크의 통신 채널을 암호화하고 있는가?	필수	
1-3. 클러스터의 민감한 데이터를 보호해야 한다.		
(1) 클러스터의 민감한 정보를 안전한 위치에 보관하고 인가되지 않은 사용자가 접근하지 못하도록 설정하고 있는가?	필수	
(2) 민감한 정보를 암호화하여 저장하고 있는가?	필수	
1-4. 감사 데이터를 생성하고 모니터링 해야 한다.		
(1) 컨테이너 운영 환경의 감사 데이터를 생성하고 모니터링하고 있는가?	필수	
(2) 감사 데이터의 임계치를 설정하고 모니터링 하고 있는가?	권고	
1-5. 주기적으로 보안 패치와 설정 점검을 수행해야 한다.		
(1) 오케스트레이터 취약점에 대한 보안 패치를 수행하고 있는가?	필수	
(2) 주기적으로 취약한 설정을 점검하고 관리하고 있는가?	필수	
1-6. 중요한 컨테이너 그룹을 식별하고, 분리된 컨테이너 자원을 할당하고 있는가?		
(1) 중요도가 높은 서비스를 식별하고, 분리된 컨테이너 자원을 할당하고 있는가?	필수	

○ 컨테이너 노드

점검항목	중요도	점검결과
2-1. 호스트 운영체제 커널의 침해에 대비해야 한다.		
(1) 가상머신 위에 호스트 운영체제와 컨테이너 엔진(런타임)을 설치하고 컨테이너를 운영하고 있는가?	필수	
(2) 컨테이너 탈출 등에 영향을 줄 수 있는 취약점을 식별하고 보안 패치를 수행하고 있는가?	필수	
2-2. 호스트 시스템 자원에 대한 접근을 제한해야 한다.		
(1) 컴퓨팅 자원이 고갈되지 않도록 컨테이너에 대한 자원 할당량을 제한하고 있는가?	필수	
(2) I/O 장치 자원이 고갈되지 않도록 컨테이너별 대역폭을 제한하고 있는가?	필수	
(3) 호스트 운영체제에 대한 컨테이너의 접근을 제한하고 있는가?	필수	
(4) 컨테이너의 호스트 네트워크 자원 사용을 제한하고 있는가?	필수	
2-3. 컨테이너를 최소 권한으로 실행해야 한다.		
(1) 특권(Privileged) 모드 컨테이너의 실행을 금지하고 있는가?	필수	
(2) 호스트와 네임스페이스를 공유하는 것을 금지하고 있는가?	필수	
(3) 오케스트레이터의 컨테이너 권한 제어 기능을 활용하고 있는가?	필수	
2-4. 관리 인터페이스에 대한 인증과 접근 제어를 강화해야 한다.		
(1) SELinux, AppArmor와 같은 강제 접근제어 기술과 Seccomp와 같은 시스템 기능 제한 메커니즘을 활용하고 있는가?	필수	
(2) 컨테이너의 비정상적인 활동을 탐지할 수 있는 보안 솔루션을 활용하고 있는가?	필수	
(3) 컨테이너의 상태에 대한 모니터링을 수행해야 하고 있는가?	필수	
2-5. 컨테이너 및 컨테이너 런타임에 대한 네트워크 접근을 제한해야 한다.		
(1) 컨테이너 런타임의 관리 인터페이스 원격 접근을 금지하고 있는가?	필수	
(2) 컨테이너 런타임의 관리 인터페이스 접근을 위한 전용 물리 NIC을 할당하거나 불가능한 경우 전용 가상 네트워크를 할당 하고 있는가?	필수	
(3) 네트워크를 통한 컨테이너 런타임의 관리 인터페이스 접속 시에는 암호화 프로토콜을 사용하고 있는가?	필수	
(4) 컨테이너 내부 접속을 위한 SSH 서버를 제거했는가?	필수	

CoreOS SSH Port 변경 요청

openmaru
오픈마루(주)

*리눅스 서버 원격 접속 서비스 **SSH**를 통한 무작위 대입 공격 여전히 위험*
8월 한 달간 총 4,436개의 IP로부터 약 65만번의 SSH 무작위 대입 시도 발생
SSH의 서비스 포트 변경, root 계정의 원격 로그인 금지, 강한 패스워드 정책 등 시행해야

https://www.boannews.com/media/view.asp?idx=109910

The 8 Most Vulnerable Ports to Check When Pentesting

https://www.makeuseof.com/vulnerable-ports-check-when-pentesting/

This kind of modification is highly discouraged and doesn't fall under tested/documented methods, keeping in mind that the Red Hat Core OS was already designed with all the possible security measures in hand.

- Red Hat Core OS는 이미 모든 가능한 보안 조치를 고려하여 설계

One such security measure is the accessibility of the RHCOS machines that is possible only through the SSH access keys generated prior to the installation, which is compatible only with the

- 설치 중에 생성한 SSH 액세스 키를 통해서만 가능한 Red Hat Core OS 시스템에 액세스 가능

https://access.redhat.com/solutions/5495101

Red Hat OpenShift OS Layer 보안

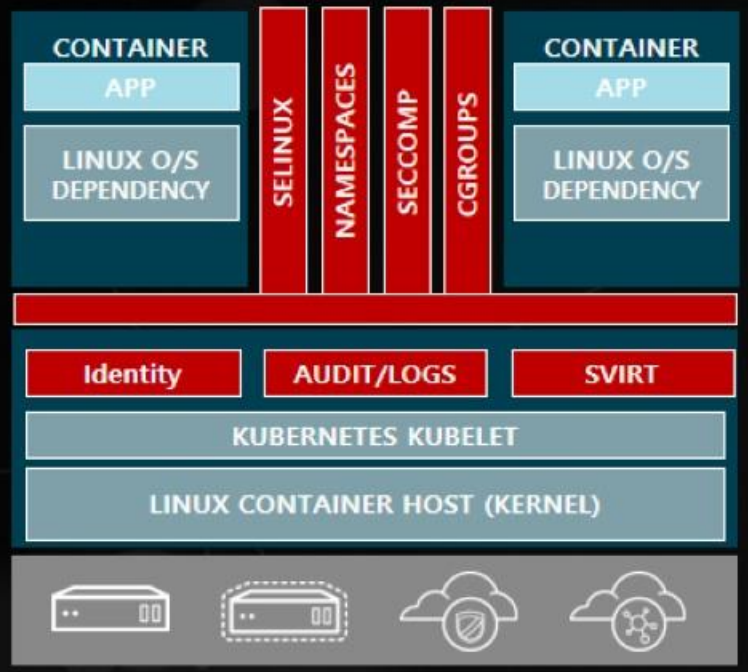
- **RHEL CoreOS로 컨테이너 호스트를 안전하게 보호**
- **Container를 기동하는 Host OS에 적합한 개념을 구현한 RHEL CoreOS**

An Ideal Container Host would be	RHEL CoreOS
Minimal	Only what's needed to run containers
Secure	Read-only & locked down
Immutable	Immutable image-based deployments & updates
Always up-to-date	OS updates are automated and transparent
Updates never break my apps	Isolates all applications as containers
Updates never break my cluster	OS components are compatible with the cluster
Supported on my infra of choice	Inherits majority of the RHEL ecosystem
Simple to configure	Installer generated configuration
Effortless to manage	Managed by Kubernetes Operators

Red Hat OpenShift OS Layer 보안 기술

openmaru
오픈마루㈜

SELINUX, NAMESPACES, SECCOMP, CGROUPS의
LINUX SECURITY 기술을 이용한 HOST OS 보호



- Security in the RHEL host applies to the container
- SELINUX and Kernel Namespaces are the one-two punch no one can beat
- Protects not only the host, but containers from each other
- RHEL CoreOS provides minimized attack surface

Machine Config Operator (MCO)

openmaru
오픈마루㈜

Provides cluster-level configuration, enables rolling upgrades, and prevents drift between new and existing nodes.

The MCO is the heart of what makes RHCOS a kube-native operating system.

Configure Kernel Arguments for the Cluster

- oc create -f 50-kargs.yaml
- oc edit mc/50-kargs

MCO can be paused to suspend operations

- Provides control for when changes can be deployed

Custom MachinePools can have inheritance

- Enables MachineConfigs to scale

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worke
r
  name: 50-kargs
spec:
  KernelArguments:
      audit=1
      audit_backlog_limit=8192
      net.ifnames.prefix=net
```

Product Manager: Ben Breard

Generally Available

Red Hat OpenShift 플랫폼 SCC 보안

openmaru
오픈마루㈜

SCC(Security Context Constraints)로 플랫폼 보안 및 접근제어

```
$ oc describe scc restricted
Name:                                   restricted
Priority:                               <none>
Access:
  Users:                                <none>
  Groups:                               system:authenticated
Settings:
  Allow Privileged:                     false
  Default Add Capabilities:             <none>
  Required Drop Capabilities:           KILL,MKNOD,SYS_CHROOT,SETUID,SETGID
  Allowed Capabilities:                 <none>
  Allowed Seccomp Profiles:             <none>
  Allowed Volume Types:                 configMap,downwardAPI,emptyDir,persistentVolumeClaim,projected,
  Allow Host Network:                   false
  Allow Host Ports:                     false
  Allow Host PID:                       false
  Allow Host IPC:                       false
  Read Only Root Filesystem:            false
  Run As User Strategy: MustRunAsRange
```

- Allow administrators to control permissions for pods
- Restricted SCC is granted to all users
- By default, no containers can run as root
- Admin can grant access to privileged SCC
- Custom SCCs can be created

Red Hat OpenShift 플랫폼 Oauth 보안

openmaru
오픈마루㈜

LDAP 및 SSO 연결로 사용자 계정 및 그룹 관리

OpenShift includes an OAuth server

- Identifies the person requesting a token, using a configured identity provider
- Determines a mapping from that identity to an OpenShift user
- Issues an OAuth access token which authenticates that user to the API

Supported Identity Providers include

- Keystone
- LDAP
- GitHub
- GitLab
- GitHub Enterprise (new with 3.11)
- Google
- OpenID Connect
- Security Support Provider Interface (SSPI) to support SSO flows on Windows (Kerberos)

Red Hat OpenShift 플랫폼 권한 관리

openmaru
오픈마루㈜

RBAC으로 전체 플랫폼의 제어 권한 관리

Role based authorization

- **Project scope & cluster scope available**
- **Matches request attributes (verb,object,etc)**
- **If no roles match, request is denied (deny by default)**
- **Operator- and user-level roles are defined by default**
- **Custom roles are supported**

For more information see: Managing RBAC in OpenShift

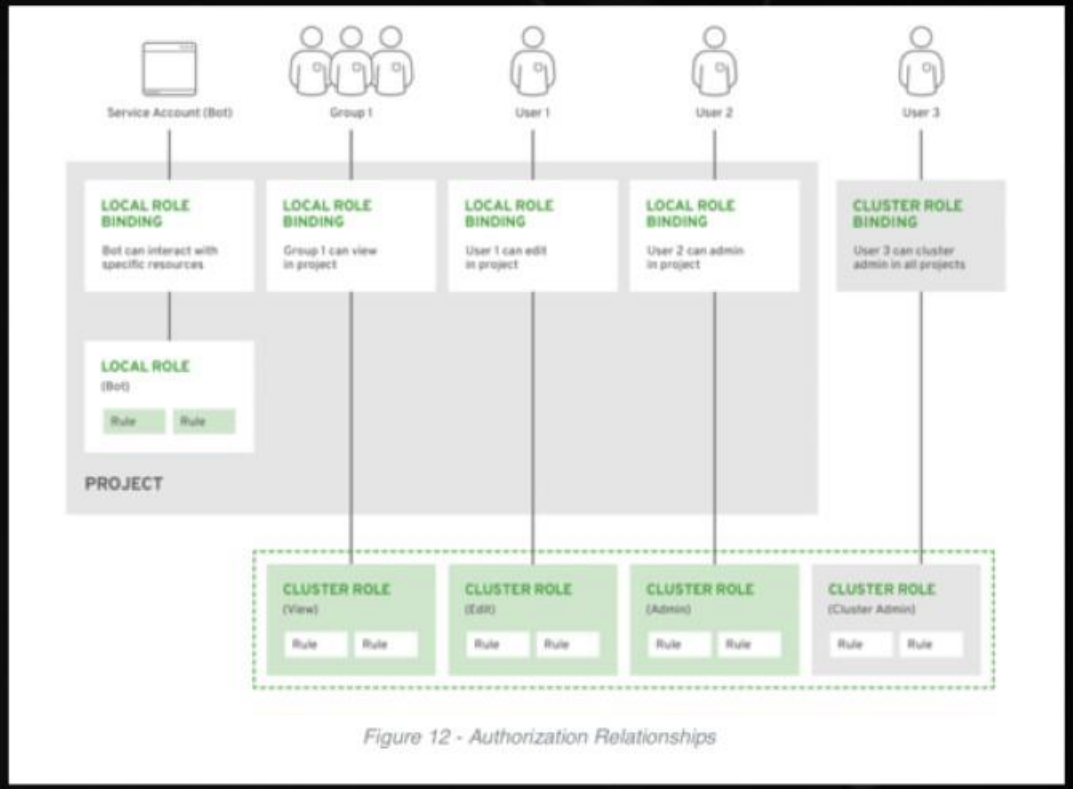


Figure 12 - Authorization Relationships

클라우드 네이티브
도입 비용 산정시 고려 사항

PaaS를 운영할 때 필요한 인프라 리스트

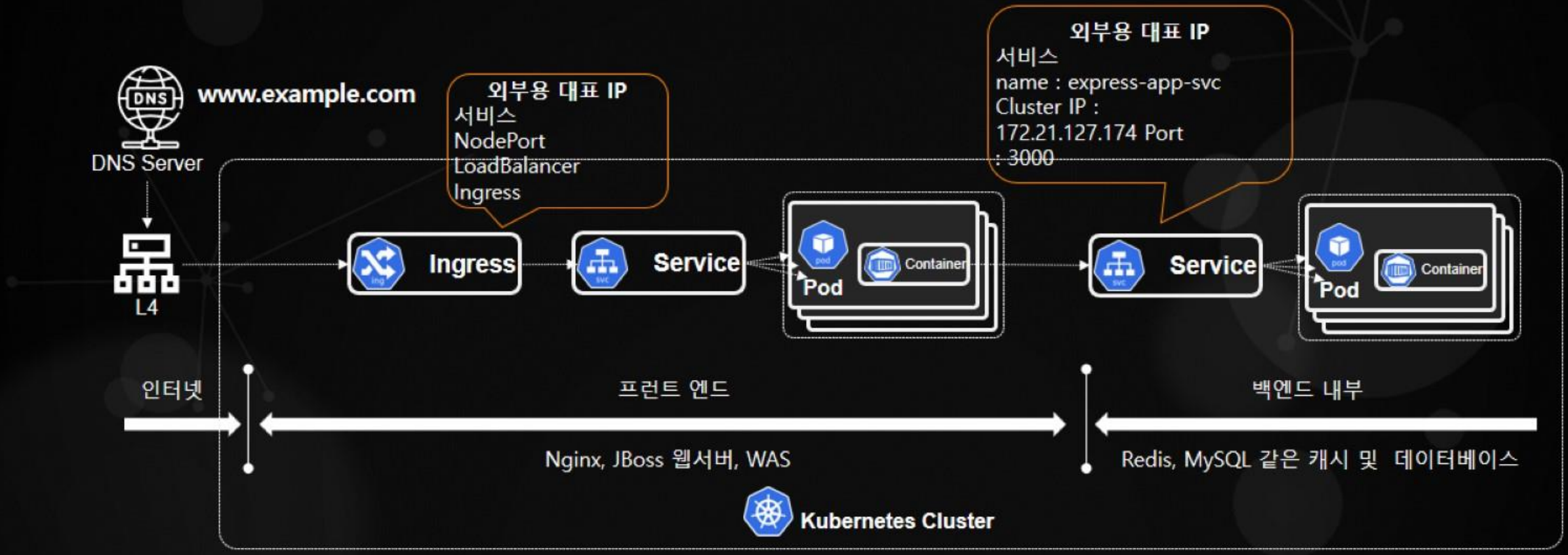
openmaru
오픈마루㈜

- 고객분들의 오해 : 플랫폼이니까 PaaS만 설치만 하면 된다?

분류	이유	역할	비고
DNS	• Cloud 에 신규서비스 추가 시 배포와 함께 서비스가 제공 되어야함 • 플랫폼 외부 네트워크에서 접속 시 도메인 기반의 애플리케이션 통신	• OpenShift Router Domain 등록	• New_App.apps.example.co.kr
L4	• 모든 Node가 Active 형태로 고가용성으로 구성됨에 따라 VIP, 헬스체크, 부하분산 필요	• Master Node, Infra Node VIP 및 로드밸런싱	
GIT	• 소스코드 저장방식의 고도화된 브랜치 전략을 가져갈 수 있음 • 소스코드만으로 컨테이너 이미지까지 빌드되는 OpenShift S2I 기능을 100% 활용	• OpenShift S2I(Source-To-Image)를 위한 소스 저장소	
Storage	• 애플리케이션들이 NAS처럼 활용 할 수 있는 공유 스토리지 • EFK, Monitoring, Image registry 등 OpenShift Infra 자원이 사용할 스토리지 공간	• 서비스가 이용할 공유 스토리지 및 OpenShift 운영 데이터 저장소	
Network	• 컨테이너 플랫폼을 구성하는 인프라 파드들과 애플리케이션 파드들도 기존 환경과는 달리 매우 많은 수가 기동되는 환경임에 따라 넓은 대역폭이 권고	• 10G 네트워크	

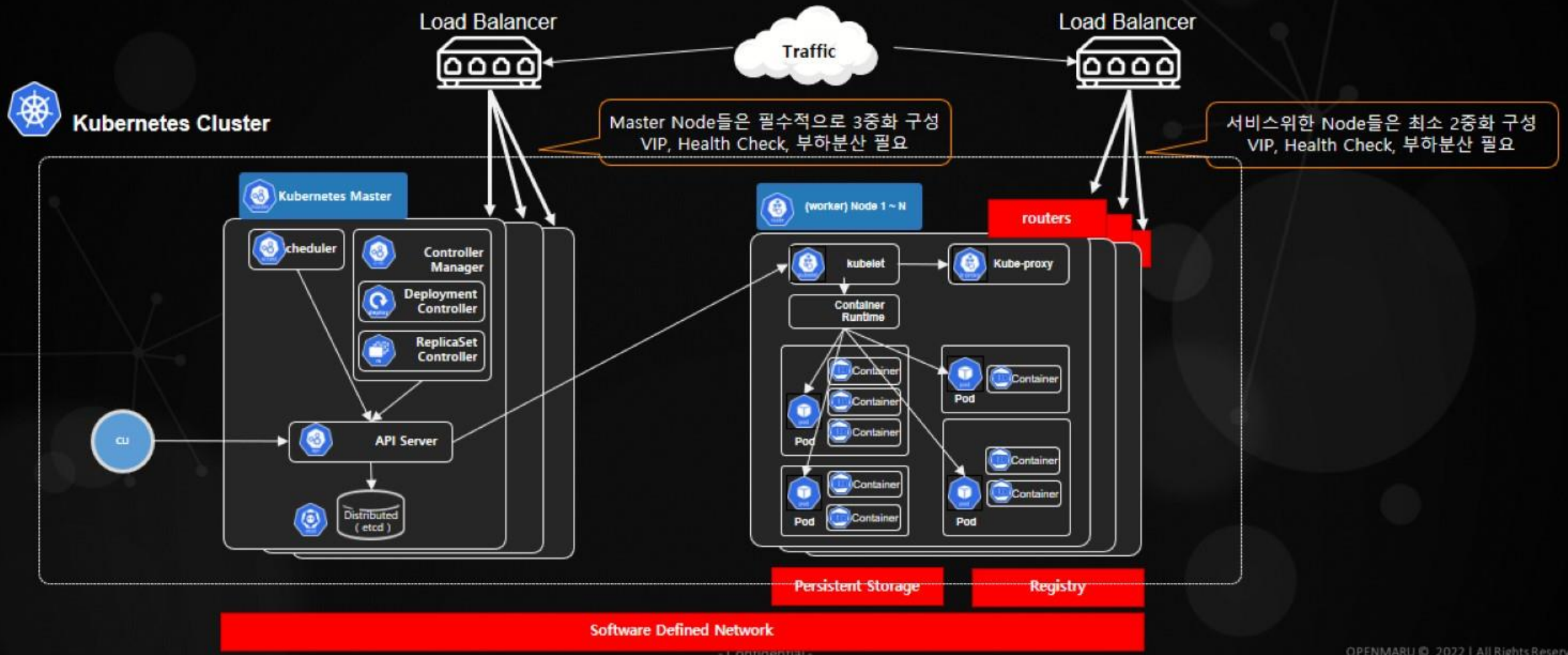
DNS는 왜 필요한가?

- 사용자들이 컨테이너 플랫폼 애플리케이션에 접속하기 위한 설정 Ingress / Route
- Domain 기반으로 OpenShift의 Pod로 들어올 수 있도록 하는 설정
- OpenShift의 애플리케이션은 Domain name 기반으로 통신



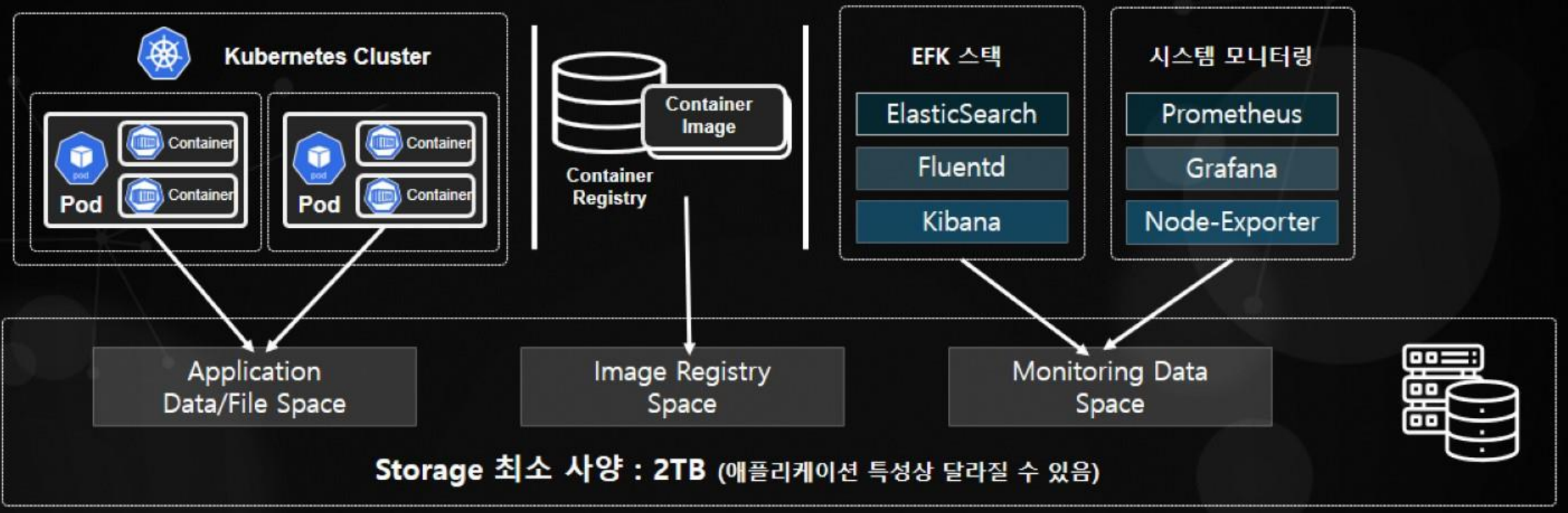
L4는 왜 필요한가?

- OpenShift의 Node들은 고가용성을 위한 다중화 구성
- OS 이중화와는 다르게 모든 Node가 Active 상태
- 그에 따라 IP를 묶어주는 VIP와 Health check, 부하분산 필요



Storage는 왜 필요한가?

- Pod(애플리케이션)에서 NAS 용도로 활용하는 공간 (Upload 파일 등)
- 큰 용량을 가지는 라이브러리 들을 저장하는 공간
- 이미지를 저장시킬 레지스트리 공간
- 로그, 시스템 메트릭 수집 데이터를 저장시킬 공간




openmaru