

# OpenShift - PaaS ( Platform As A Service )



TensorFlow



theano



torch



hadoop



cassandra



APACHE Spark



mongoDB



STORM



kubernetes



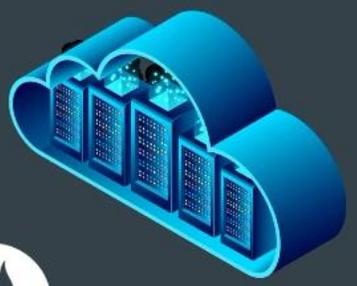
fluentd



Prometheus



openstack



HYPERLEDGER



HYPERLEDGER ARIES



HYPERLEDGER BURROW



HYPERLEDGER BESU



HYPERLEDGER CALIPER



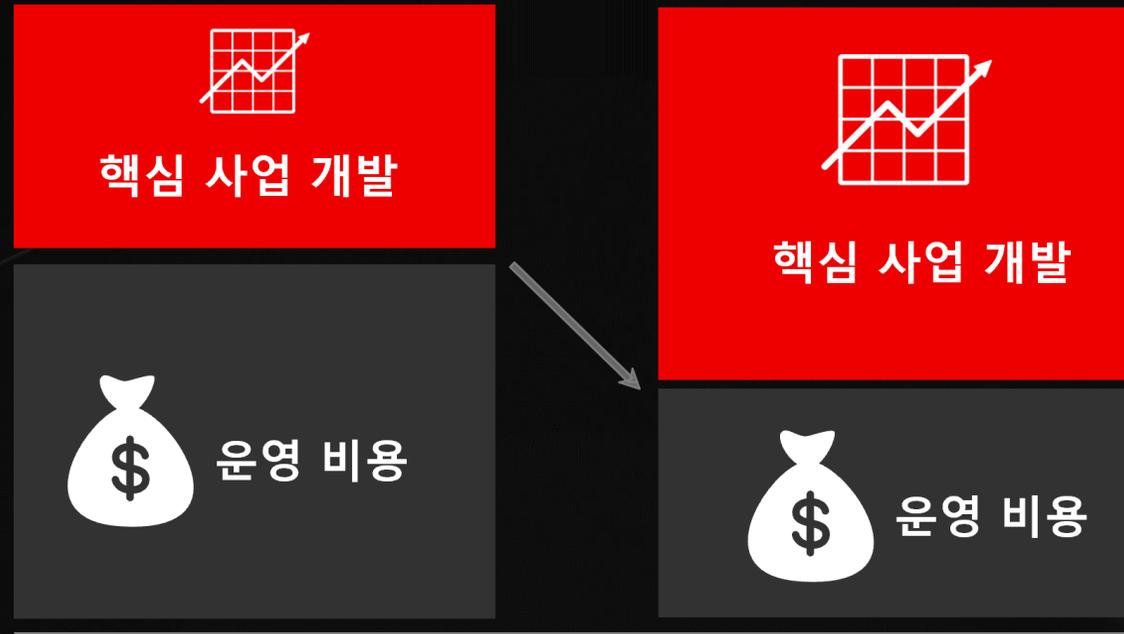
# 소프트웨어 시대

- 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



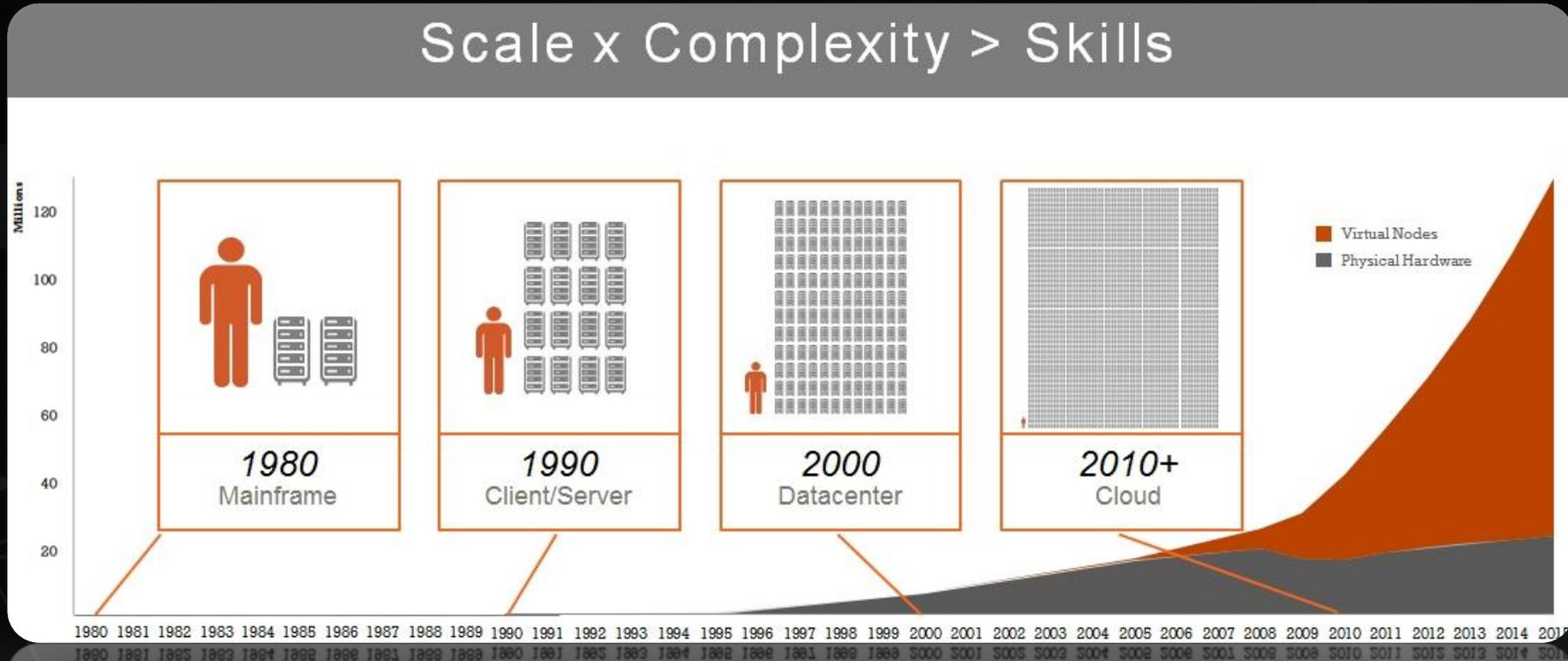
# 클라우드 네이티브 기술을 통한 사업 비용 절감

- 매니지드 서비스로 운영 비용을 줄이고 귀중한 인적 자원을 핵심 사업의 개발에 집중



관리에 대한 기대

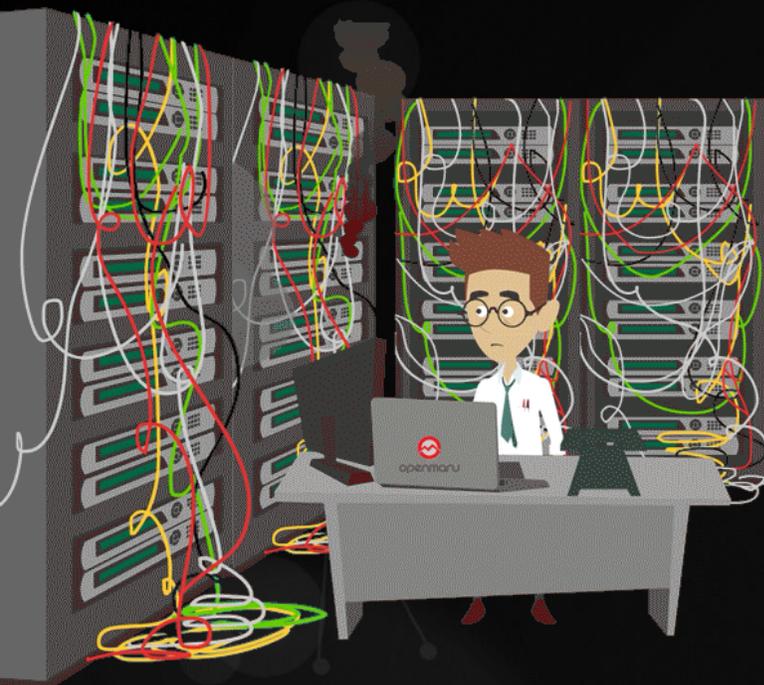
# Increasing scale and complexity means we need admin automation



Opscode gets more venture dough for its Chef

From - <http://goo.gl/dLcjS>

# 시스템 비대화로 작업 폭증과 인력 부족 어떻게 할까요?



장애의 65 %는 Human Error이며, 시스템 복잡도와 난이도 증가

시스템 운용 업무의 45 %는 정기적으로 수행해야 하는 반복 작업

운영 효율화를 통한 비용 절감의 요구



시스템의 대규모화



높은 수준의 엔지니어 부족



지속적인 시스템 통합 요구



동일한 작업 반복



운영 품질 향상



운영 비용 (TCO) 절감 요구

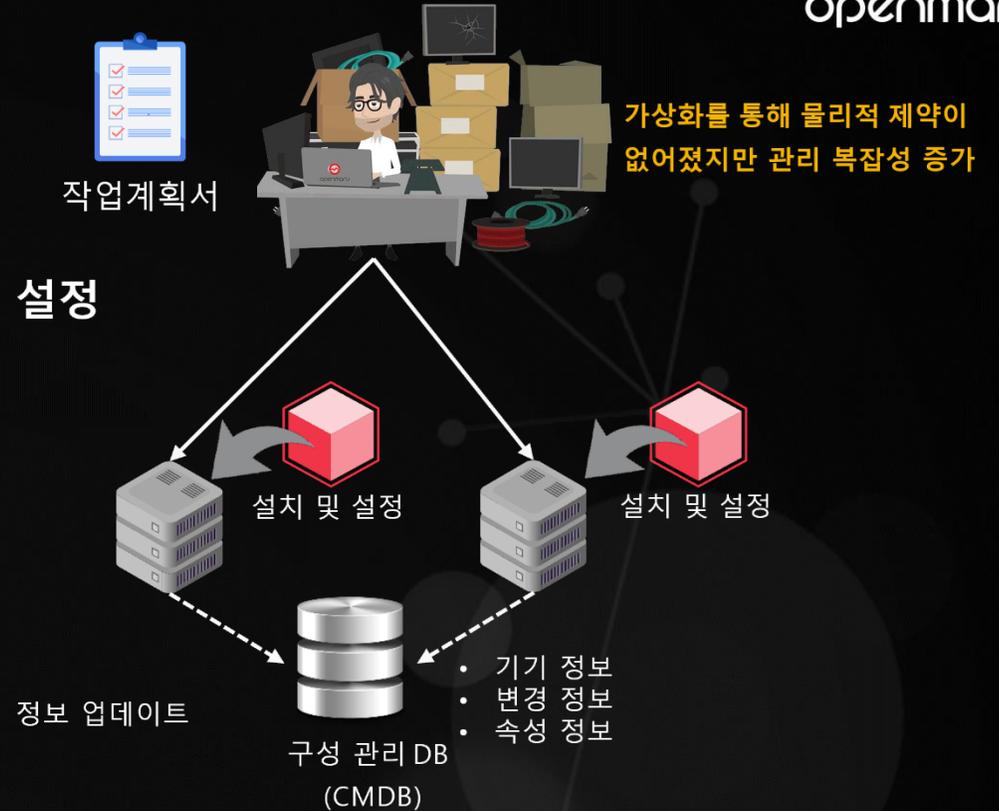
업무 확대와 관련 데이터양의 비약적인 증가

가상화, 클라우드 등 다양한 운영 환경의 증가와 관리 효율화 요구

운영 품질에 대한 지속적인 향상 요청

# 기존의 구성 관리 이슈

- 기존 온-프레미스 환경에서
  - 엔지니어가 복잡한 인프라 구축
  - 구축 후 만료될 때까지 장기적 운영
  - 미들웨어 클러스터 구축, 수백 대의 OS 초기 설치 및 설정
  - 운영 시 변경 관리와 CMDB 수동 업데이트 등



- 클라우드 환경으로 전환하면서
  - 인프라 자원의 라이프 사이클이 짧아짐
  - 구축과 운영 비용이 급증



- OpenShift 을 적용하여 민첩성 높은 서비스를 제공



## 작업 공수 절감

기존 수동으로 해왔 던 작업을 자동화하여 작업 공정 및 납기 단축

## 운영 품질 향상

관리자의 개입을 최소로 자동화하여 작업 품질을 균일하게 유지

## 시스템 운영의 표준화 촉진

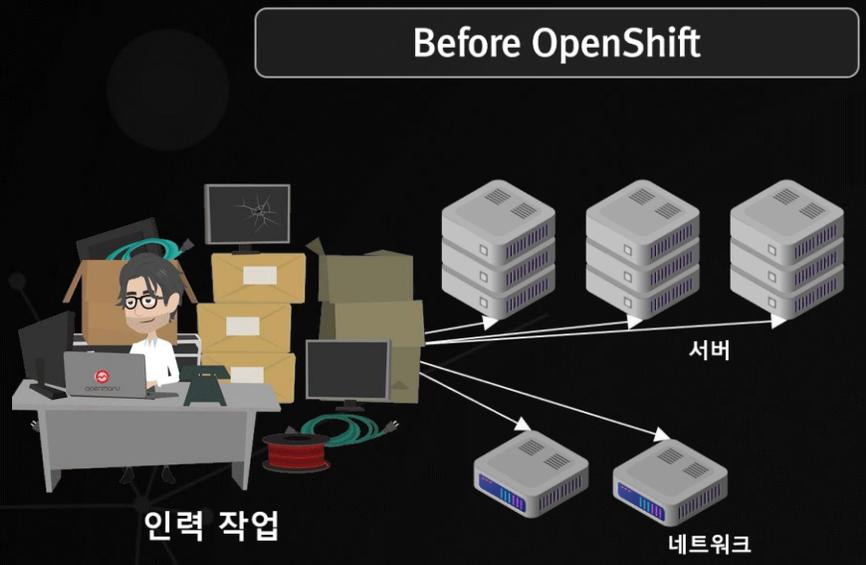
- 자동화 및 버전 관리 함으로써 시스템 운영 정책 및 업무 표준화

## 작업 통제 강화

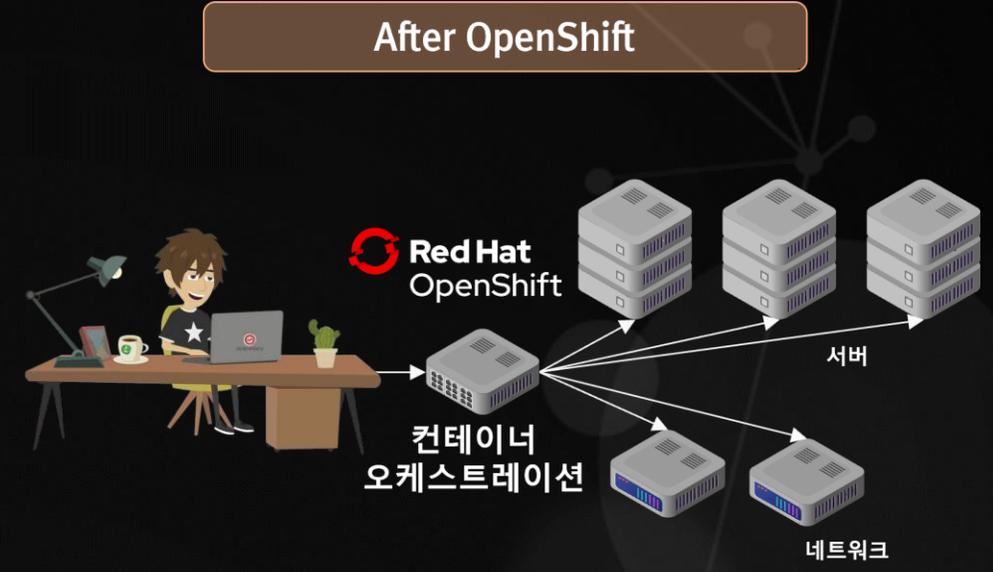
작업 작업을 자동화함으로써 내부 통제 및 보안 측면에서의 효과를 기대

# OpenShift 을 통한 IT 인프라 운영 자동화

- IT 인프라의 대규모화, 고도화에 따라 IT 장비에 대한 환경설정 및 정보 취합이 복잡하고 어려움
- 작업 계획시간과 현장 작업 시간의 증가와 휴먼 에러의 증가



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



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



# 컨테이너 도입 패턴



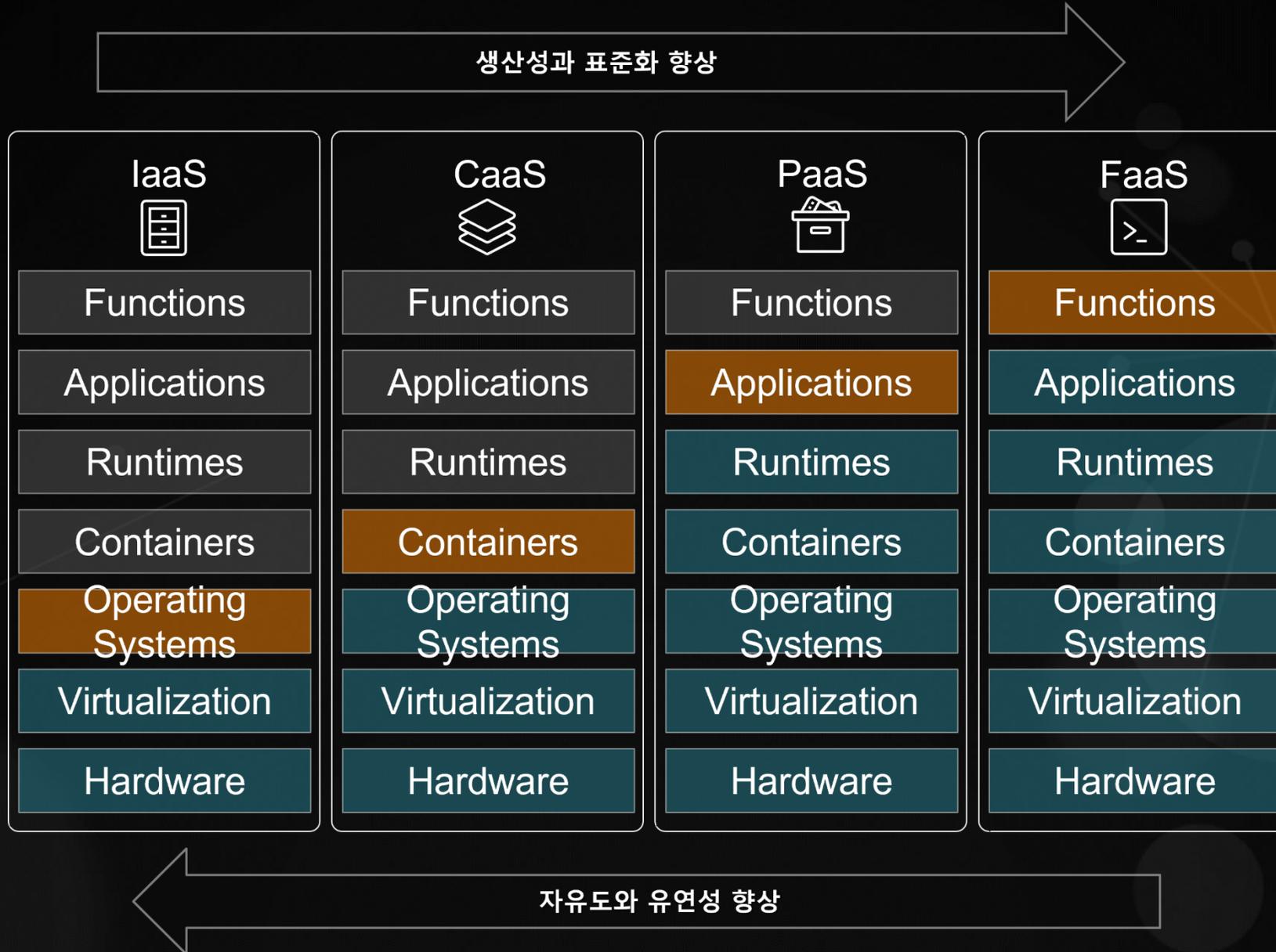
Container Opportunity

# 컨테이너 도입 패턴



분류	패턴1	패턴2	패턴3	패턴4
	컨테이너 플랫폼	클라우드 네이티브	하이브리드 클라우드	비즈니스 혁신
사례	<ul style="list-style-type: none"> <li>• 컨테이너 기술과 DevOps 를 활용한 애플리케이션 배포 효율화</li> </ul>	<ul style="list-style-type: none"> <li>• 기존 애플리케이션의 쇄신과 새로운 애플리케이션 개발</li> </ul>	<ul style="list-style-type: none"> <li>• 개발 팀을 위한 IaaS/PaaS 구축</li> </ul>	<ul style="list-style-type: none"> <li>• 새로운 비즈니스 모델에 따른 신규 사업을 위한 애플리케이션, 서비스 구축</li> </ul>
동기	<ul style="list-style-type: none"> <li>• 클라우드 인프라로 필요한 기능을 올인원으로 제공</li> <li>• 기업에서 사실상의 표준이 된 Kubernetes 의한 컨테이너의 오케스트레이션</li> </ul>	<ul style="list-style-type: none"> <li>• 마이크로 서비스</li> <li>• 경량 프레임워크와 플랫폼 ( Spring Boot, Wildfly Swarm 등)</li> <li>• API 서비스</li> </ul>	<ul style="list-style-type: none"> <li>• 신뢰할 수 있는 플랫폼</li> <li>• 자원의 효율적인 활용</li> </ul>	<ul style="list-style-type: none"> <li>• 조직 변경을 포함한 이노베이션</li> </ul>
장점	<ul style="list-style-type: none"> <li>• IaaS 공급자에 독립적</li> </ul>	<ul style="list-style-type: none"> <li>• 확장성</li> </ul>	<ul style="list-style-type: none"> <li>• 사내 / 공용 클라우드에 걸쳐 유연한 환경</li> </ul>	<ul style="list-style-type: none"> <li>• 새로운 비즈니스 기회 창출</li> </ul>
대상	<ul style="list-style-type: none"> <li>• CTO</li> <li>• 엔터프라이즈 아키텍트</li> <li>• 운용 / DevOps 팀</li> </ul>	<ul style="list-style-type: none"> <li>• 애플리케이션 아키텍트</li> <li>• 애플리케이션 개발 리더</li> </ul>	<ul style="list-style-type: none"> <li>• 엔터프라이즈 아키텍트</li> <li>• CTO</li> <li>• 인프라 및 운영 팀</li> </ul>	<ul style="list-style-type: none"> <li>• CTO, CIO</li> <li>• Chief Digital Officer</li> <li>• 비즈니스 선도 부서</li> </ul>

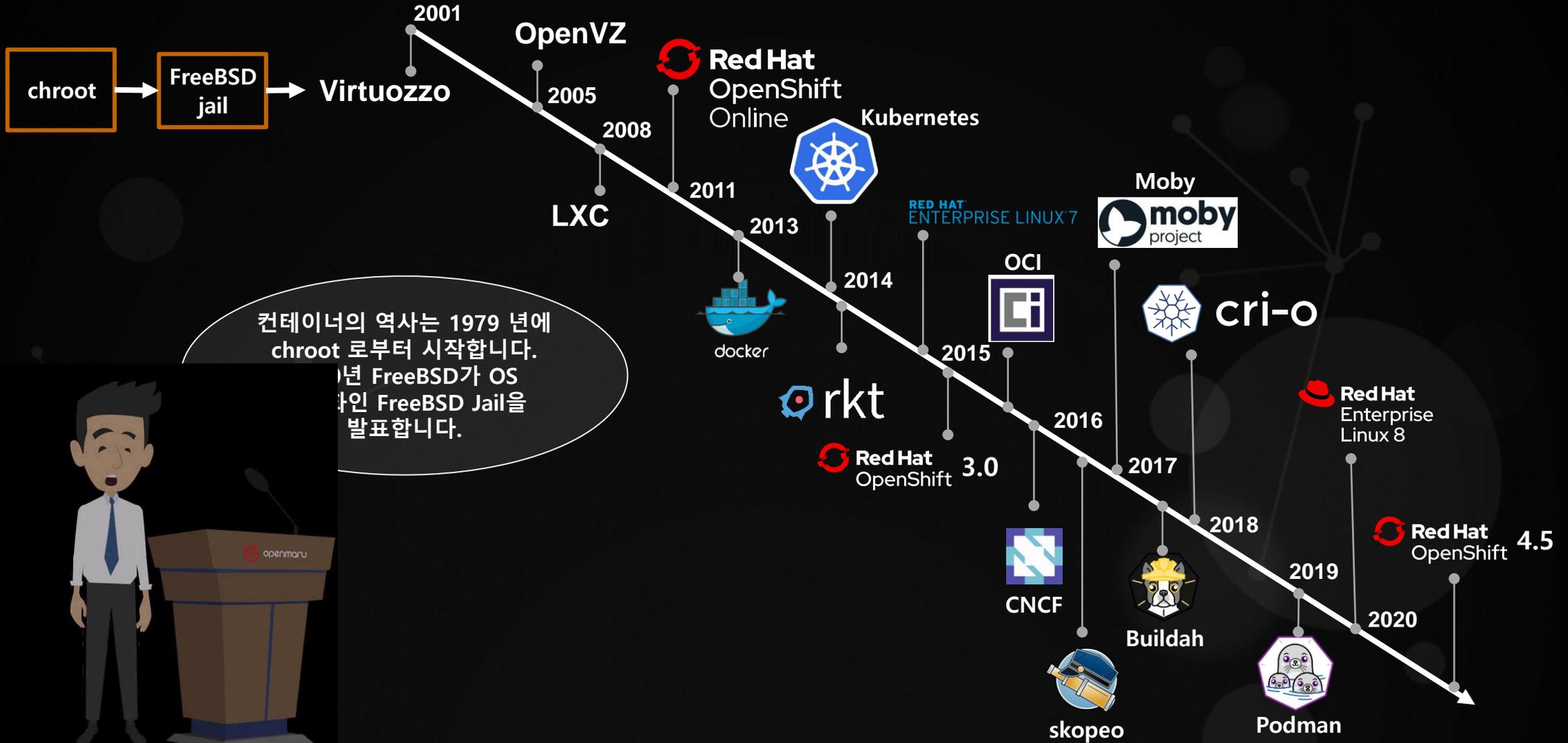
# Architecture Battle: PaaS vs. FaaS



Platform As A Service

# Kubernetes vs. OpenShift

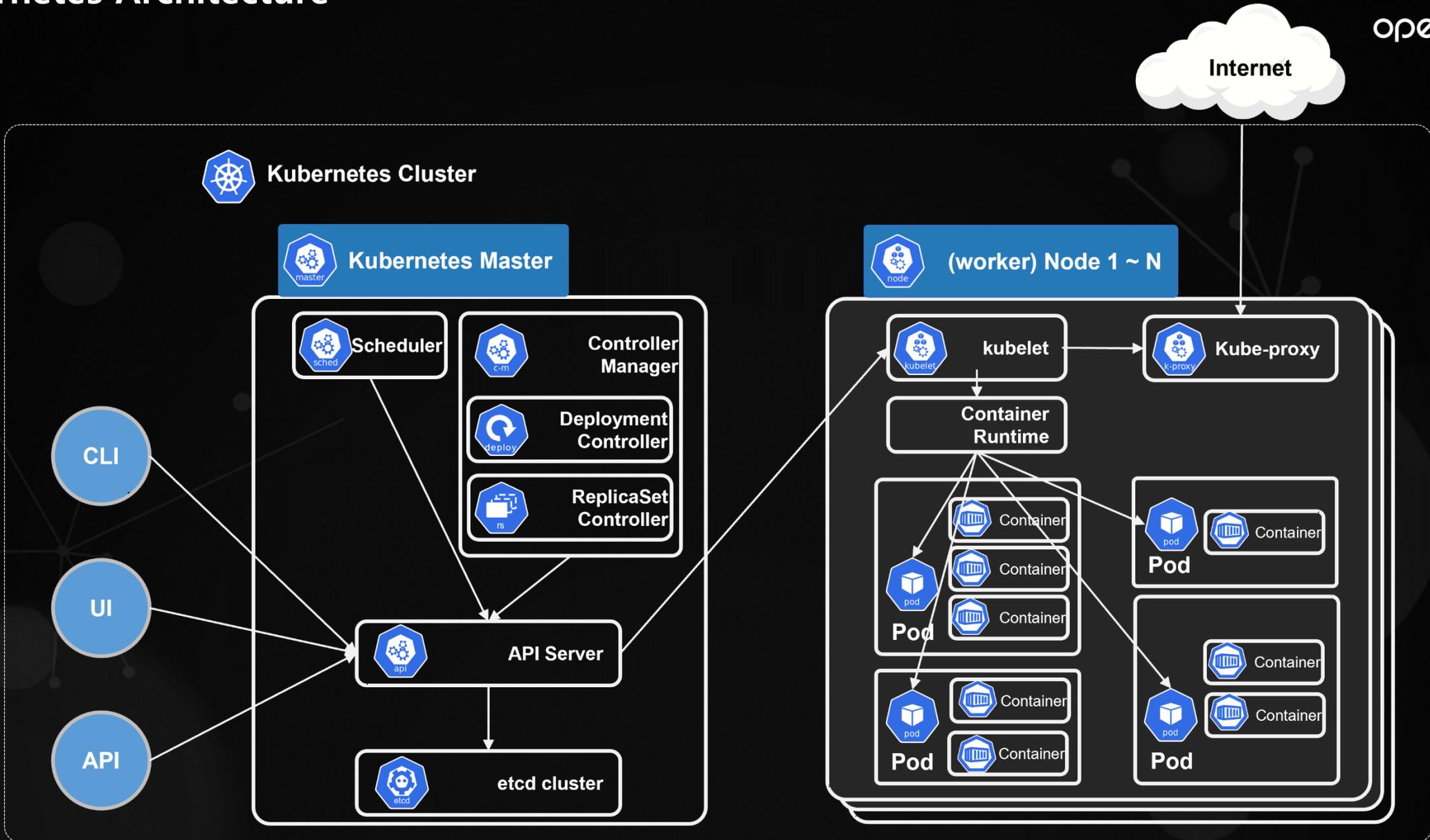
# 컨테이너 기술의 역사



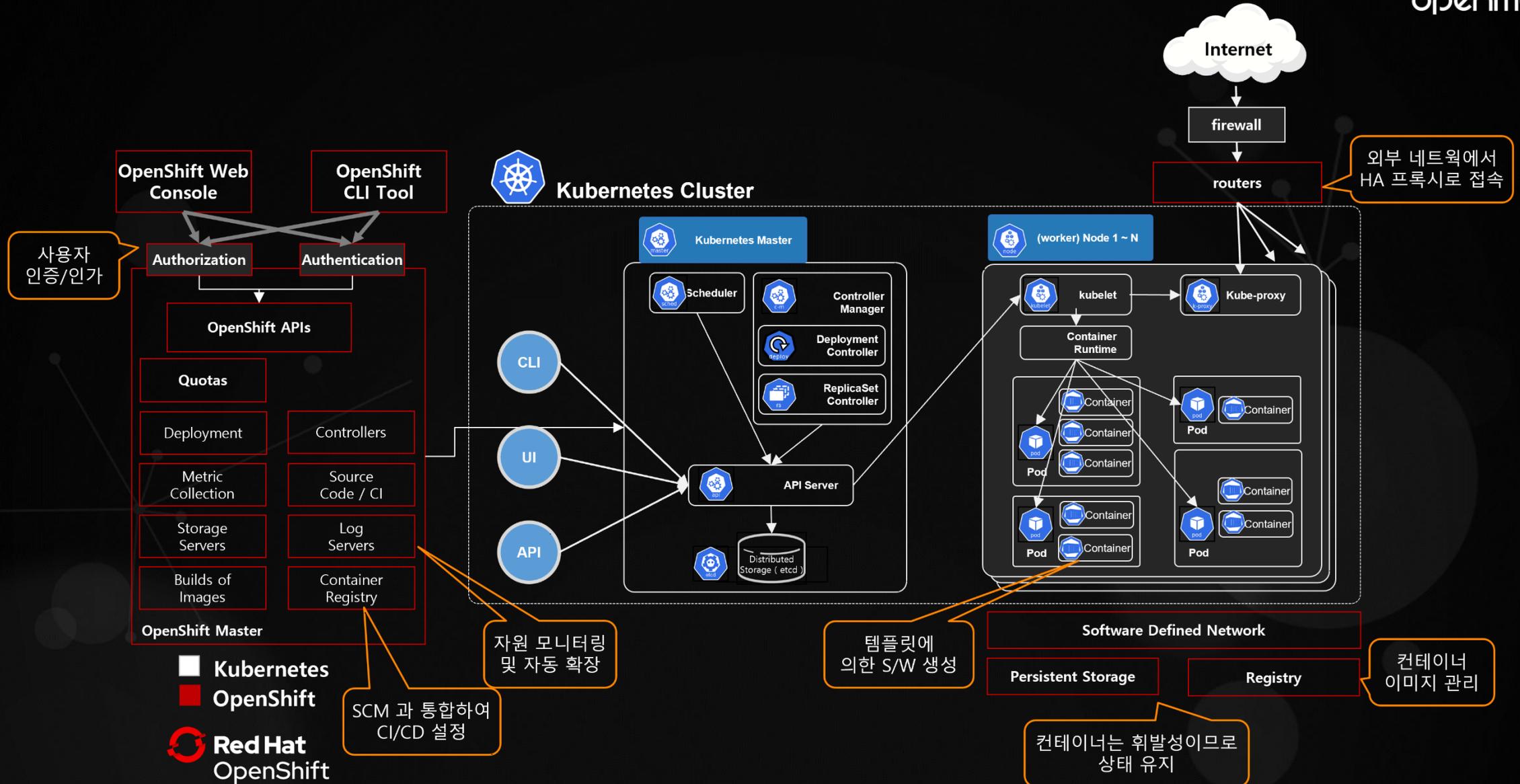
컨테이너의 역사는 1979 년에 chroot로부터 시작합니다. 1999년 FreeBSD가 OS 라인 FreeBSD Jail을 발표합니다.



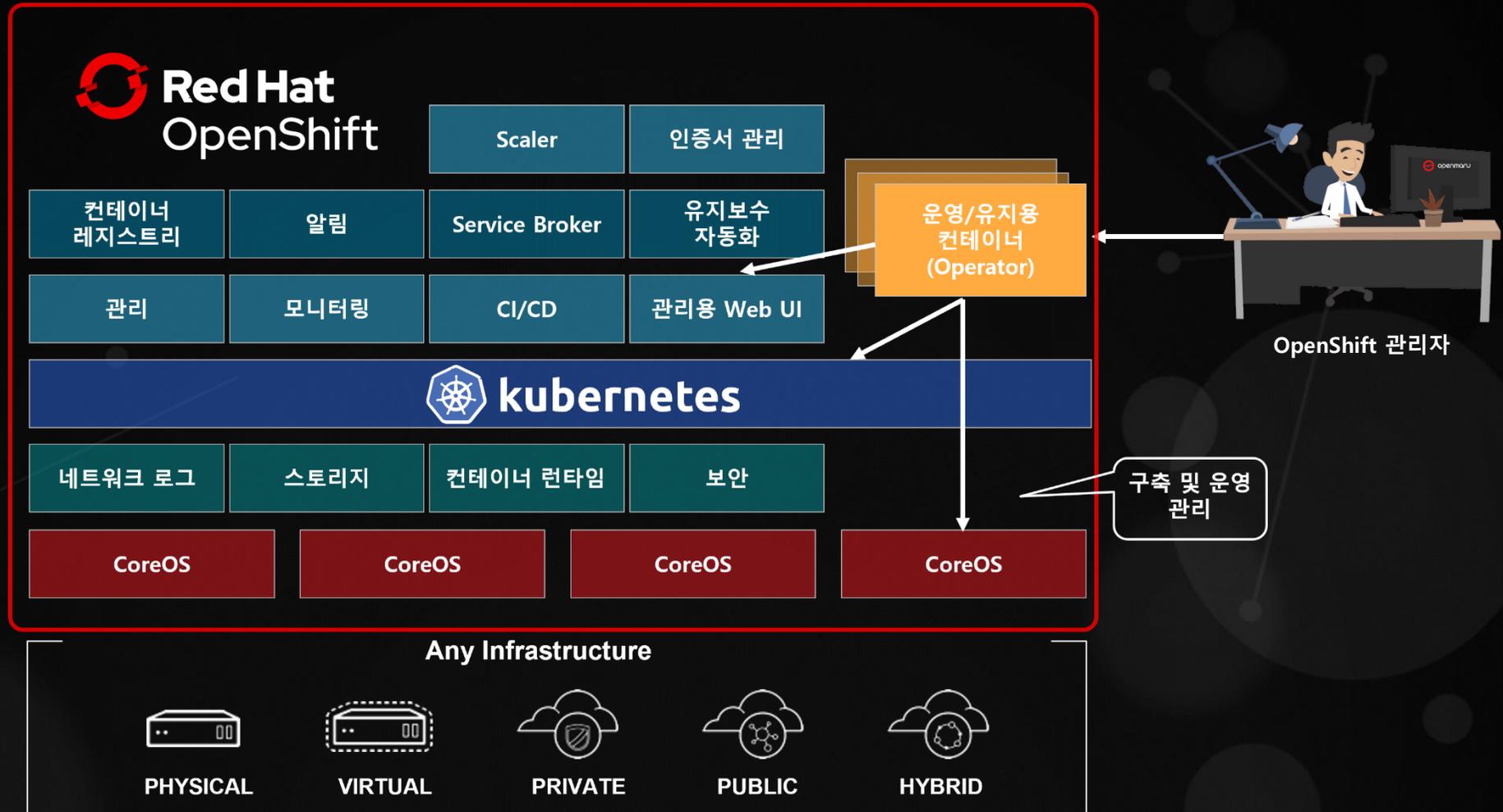
# Kubernetes Architecture



# OpenShift Architecture



# 컨테이너 플랫폼 구축 및 관리 - OpenShift4



# KUBERNETES 를 제대로 운영하기 어려운 이유는?

- Kubernetes 훌륭한 기초 기술이지만, 스토리지, 네트워킹, 보안, 애플리케이션 프레임 워크 등을 통합하고 이를 분기별로 갱신하는 것은 큰 부담입니다

- - Ashesh Badani, Red Hat

## Why running your own Kubernetes deployment could be a terrible idea

Kubernetes is hard, but becomes doubly so when you take on the burden of supporting this fast-moving project.

By Matt Asay  | June 21, 2018, 11:23 AM PST

<https://www.techrepublic.com/article/why-running-your-own-kubernetes-deployment-could-be-a-terrible-idea/>

# DIY Kubernetes 만으로는 구축이 어려운 점

- Kubernetes 는 기본적인 컨테이너 오케스트레이션을 제공하지만 실제 운영이 가능 하려면 고객 또는 3rd-party 가 구성 통합 운영 및 지원하는 추가적인 인프라스트럭처 필요

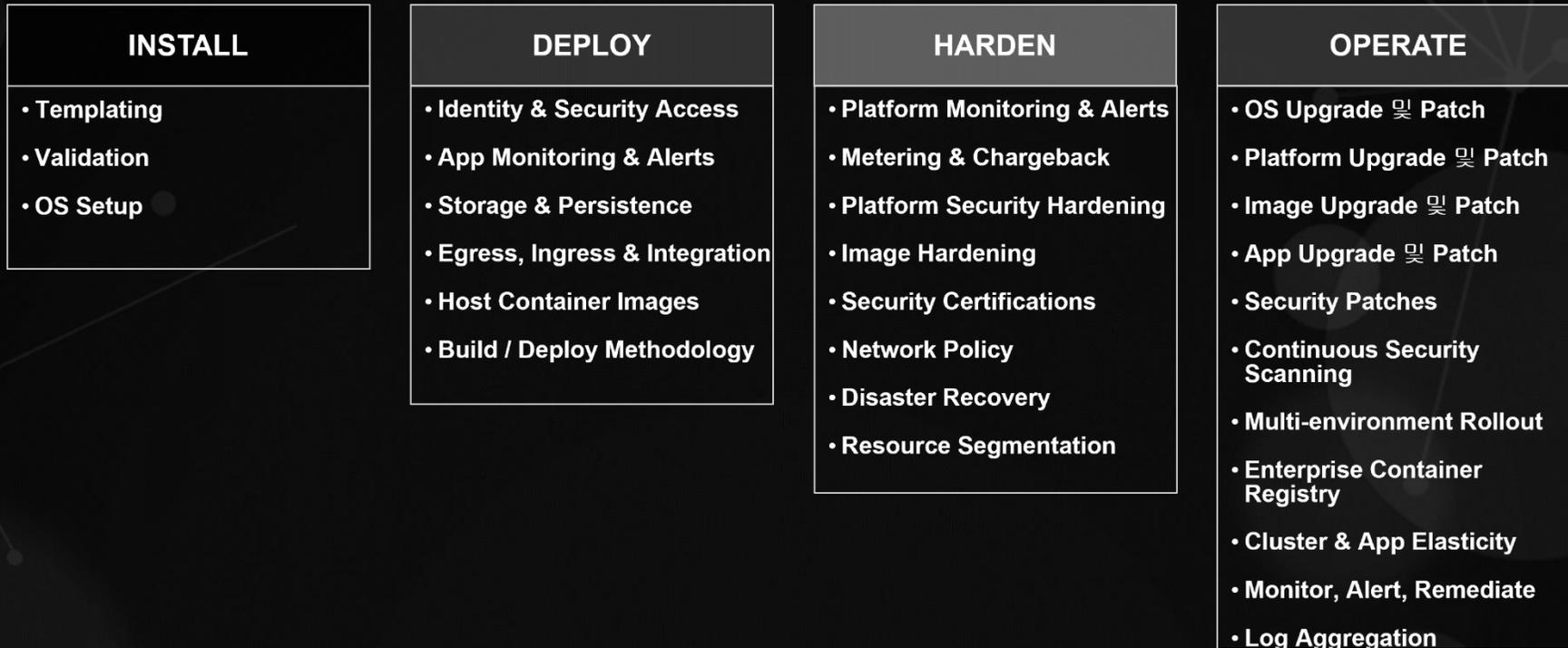
1. Linux 운영 체제 필요
2. 컨테이너 런타임 ( CRI-O , Containerd , Docker 등) 이 필요
3. 이미지 레지스트리가 필요
4. SDN ( Software-Defined Network) 이 필요
5. 로드밸런서 및 라우팅 필요
6. 로그 관리가 필요
7. 컨테이너 자원 사용률 측정 항목 및 모니터링 필요
8. 컨테이너 빌드 및 배포 (+자동화)

**OpenShift 는 플랫폼의 일부로 완벽하게 통합되고 완벽하게 테스트 된 모든 구성 요소를 포함**

# KUBERNETES 를 제대로 운영하기 어려운 이유는?

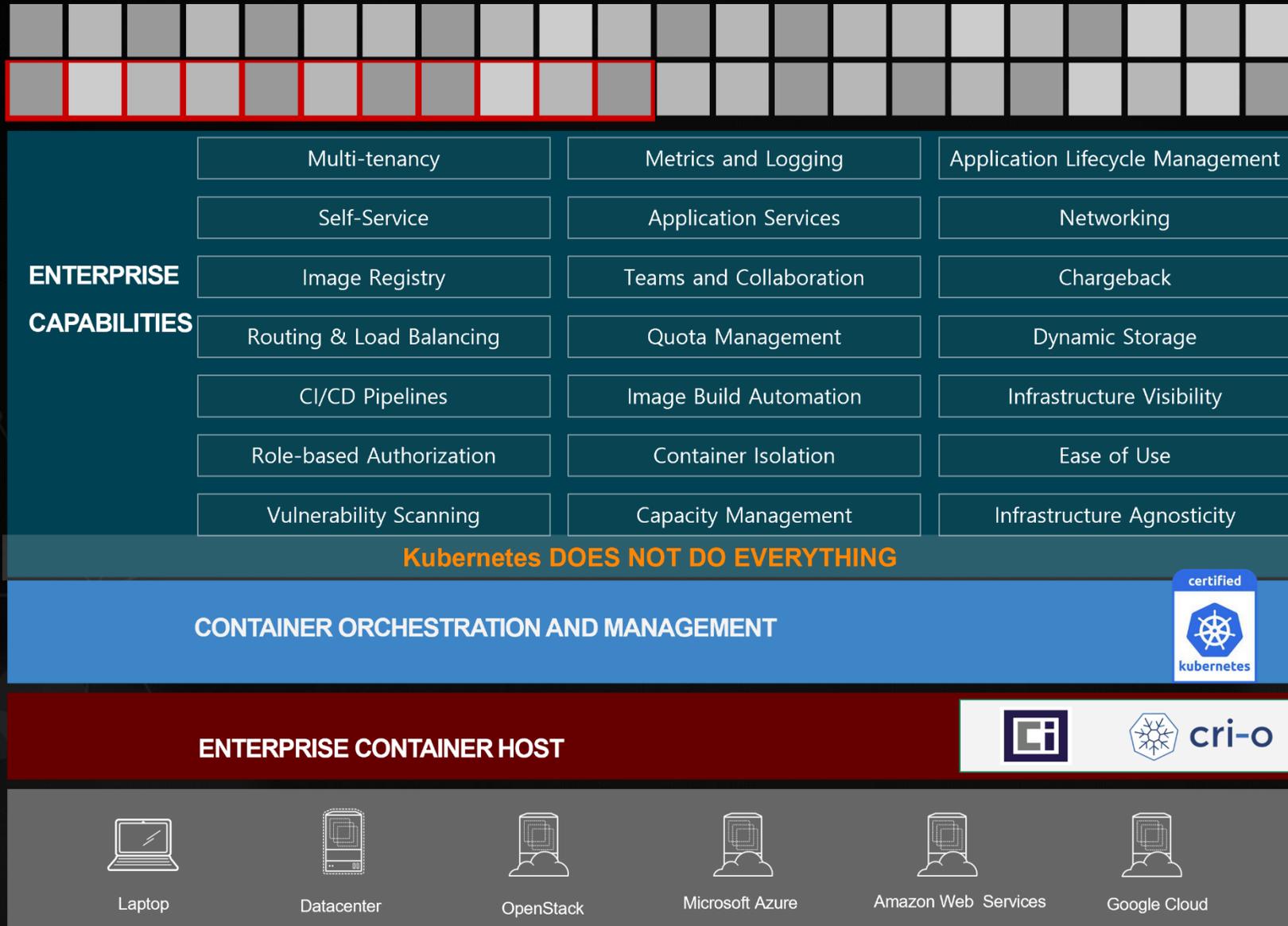
Kubernetes 사용자 중 **75 %** 는

구축과 운영의 복잡성으로 도입하기 어렵다고 함



Source : The New Stack, The State of the Kubernetes Ecosystem, August 2017

# OpenShift 4, the smarter Kubernetes platform



ANY OCI COMPLIANT CONTAINER



**Red Hat OpenShift**

+ a wide range of value added products

ANY INFRASTRUCTURE

# Kubernetes와 OpenShift의 차이점



항목	 kubernetes	 Red Hat OpenShift
제품	오픈 소스 프로젝트이자 프레임워크, 제품은 아님	Red Hat이 개발하는 제품으로 OpenShift는 다양한 변형을 가진 제품으로 OKD라고 부르는 오픈소스 버전도 존재
설치 플랫폼	거의 모든 Linux에서 설치 가능	Red Hat Enterprise Linux 혹은 CoreOS를 지원함
주요 클라우드 플랫폼	Google GCP의 GKE, Amazon AWS의 EKS 및 Microsoft Azure의 AKS에서 사용 가능	AWS, GKE, Azure, OpenStack, VMWare, Bare Metal 등 다양한 환경 지원
보안 & 인증	잘 정의되어 있지만 OpenShift만큼 엄격하지는 않음	보다 엄격한 보안 정책 및 인증 모델
Releases	거의 일 년에 4번 릴리즈	1년에 4번 릴리즈
Deployment	컨트롤러를 통해 구현된 배포 개체	로직을 구현하는 DeploymentConfig 객체를 제공
업데이트	배포에서 한 번에 여러 개의 동시 업데이트 지원	DeploymentConfig가 동시 업데이트를 지원하지 않음
컨테이너 이미지 관리	컨테이너 이미지 관리를 위한 특정 기능이 없음	OpenShift의 고유한 기능인 ImageStream으로 컨테이너 이미지를 쉽게 관리
CI / CD	CI/CD는 Jenkins와 함께 사용할 수 있지만 툴에 통합되어 있지 않음	OpenShift에 포함된 Jenkins로 원활한 CI/CD 통합 가능
초기 설치 및 구성	추가 구성 요소가 있어 초기 설치 및 구성이 다소 어려움	최소한의 추가 기능으로 완벽한 클러스터 솔루션으로 작동하므로 간편하게 설치/구성 가능
Enterprise Usage	Kubernetes는 기업 사용을 위한 특별한 제공은 없음	OpenShift는 기업을 염두에 두고 설계되었으며 이를 위한 많은 구성요소를 제공하고 있음
네트워크	타사 네트워크 플러그인에서 사용할 수 있는 인터페이스 제공	3가지 다양한 플러그인을 제공하는 네이티브 네트워킹 솔루션인 OpenvSwitch가 있음
웹 사용자 인터페이스	대시보드 별도 설치해야 함, 로그인 페이지 수동 생성 필요	관리 웹 애플리케이션 기본 제공, 다양한 리소스를 콘솔에서 손쉽게 생성 가능

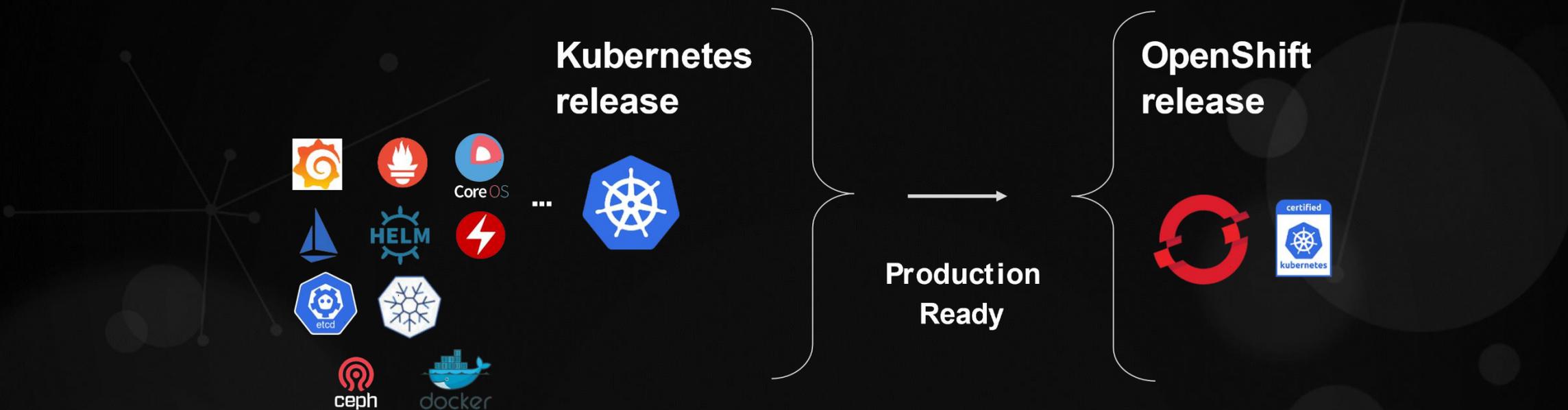
# Kubernetes vs. OpenShift Comparison

분류	Kubernetes	OpenShift
Developers	CNCF ( Cloud Native Computing Foundation )	Red Hat
Programming Language	Go	Go , Angular JS
Category	Cluster Management Software	Cloud Computing , PaaS
License	Apache License 2.0	Apache License 2.0
Released in	2014	2011

<https://hackernoon.com/kubernetes-vs-openshift-a-detailed-comparison-7r3z53zlv>

# OpenShift is trusted Enterprise Kubernetes

- Hundreds of defect and performance fixes
- 200+ validated integrations
- Certified container ecosystem
- 9-year enterprise life-cycle management
- Red Hat is a leading Kubernetes contributor since day



# 기업에서 필요한 Kubernetes 기술지원

## 장기 수명주기

Predictable and long lifecycle

예측 가능한 라이프 사이클을 제시하여  
운영하는 애플리케이션과  
비즈니스에 필요한 장기 지원 체계 제공

## SLA 와 기술 지원

SLA and support

장애에 대한 응답 및 복구 할  
수 있는 기한에 따라 SLA 을  
지원하며, 벤더를 통한 명확한  
지원 체계 제공

## 교육

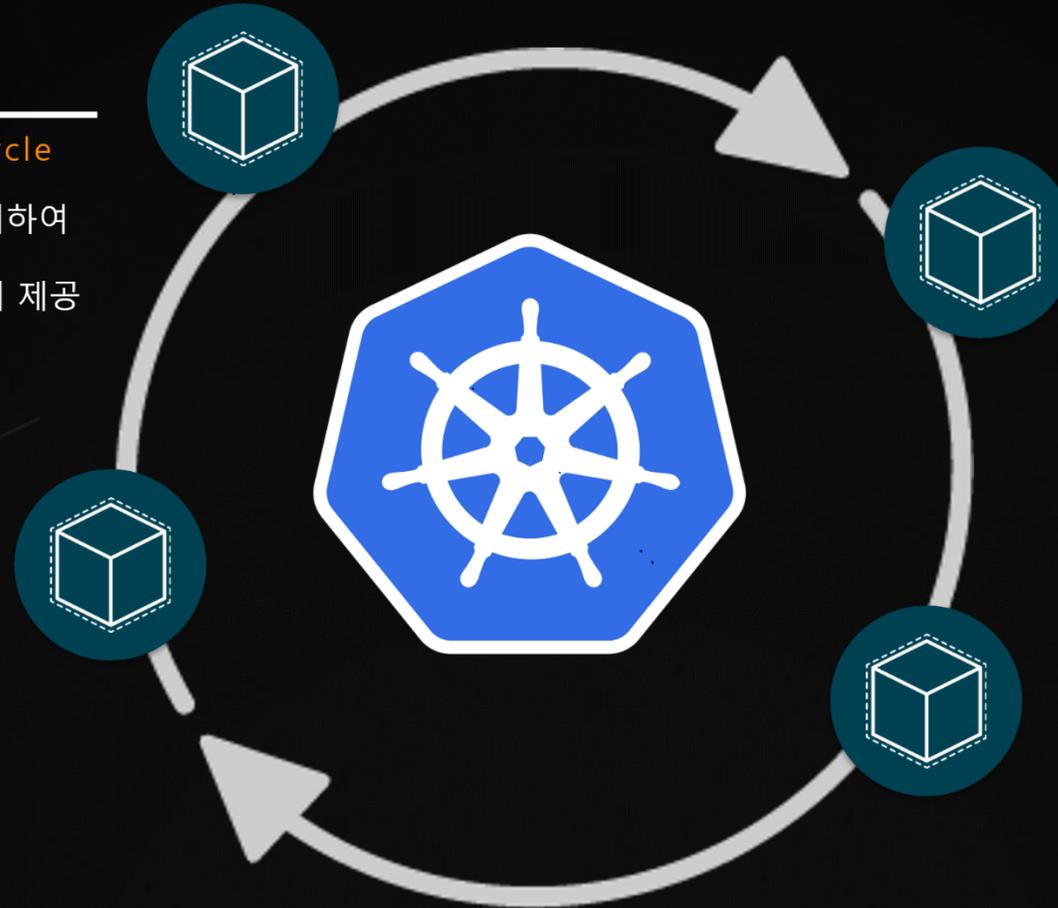
Training

장기적인 지원 뿐만 아니라 제품 교육  
과 자격증 제도를 제공하여 기술 내재  
화를 통한 서비스의 지속적인 운영을  
지원

## 파트너 인증

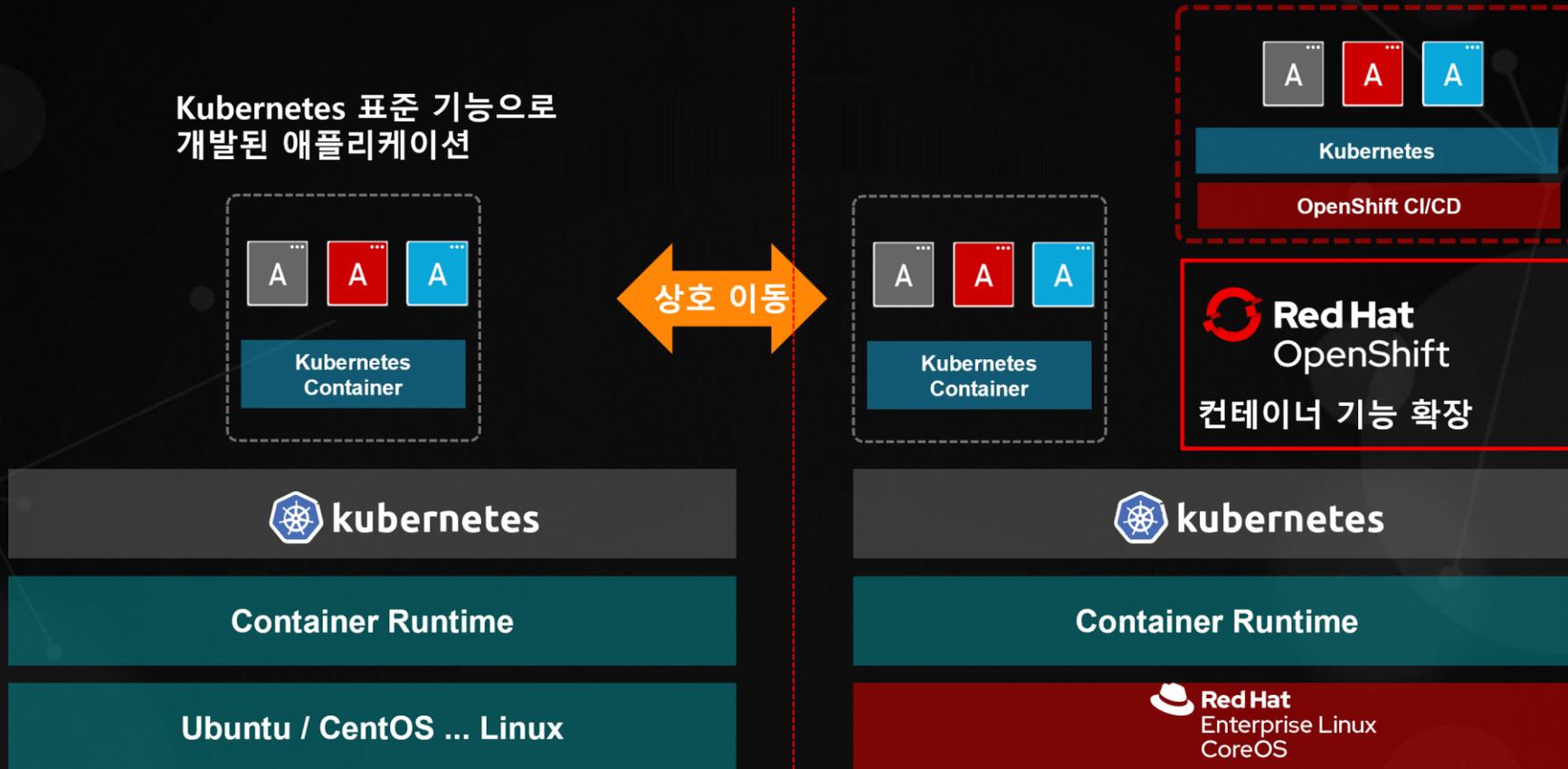
Certification

ISV 가 제공하는 3rd party 소프트웨어를 포함하  
여 쿠버네티스 상에서  
동작을 확인하고 비즈니스에 중요한 워크로드에  
적합한 지 보장



# Kubernetes와 OpenShift

- OpenShift 은 Kubernetes 과의 호환성을 유지하면서 CI/CD 기능을 강화
- 자동화된 이미지 빌드 배포 지원



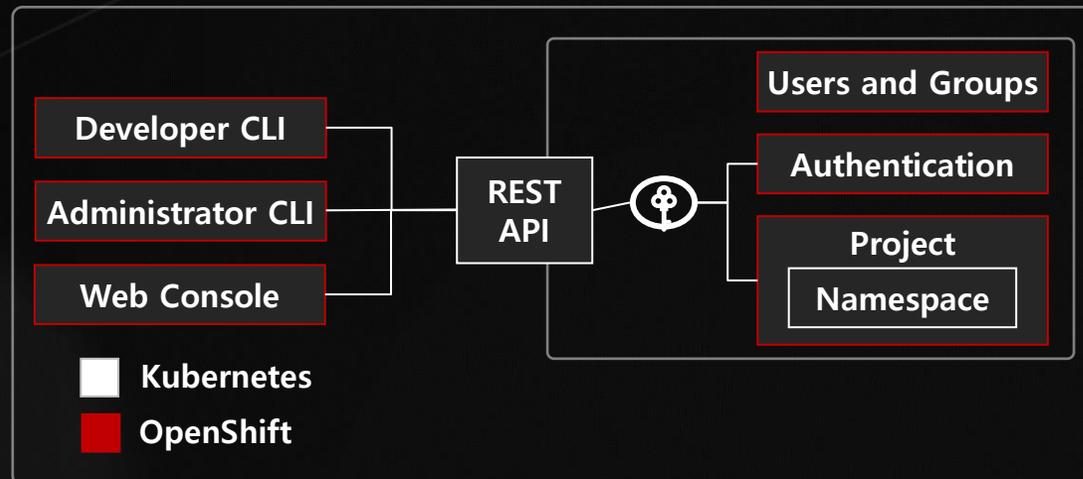
# OpenShift Project 와 Kubernetes Namespace

- **Namespace**

- Kubernetes의 기본 그룹화 개념으로 클러스터 리소스를 여러 용도로 나누는 방법
- Kubernetes의 네임 스페이스 사이에는 보안이 없이 사용자는 모든 다른 네임 스페이스와 여기에 정의 된 리소스에 접근

- **Project**

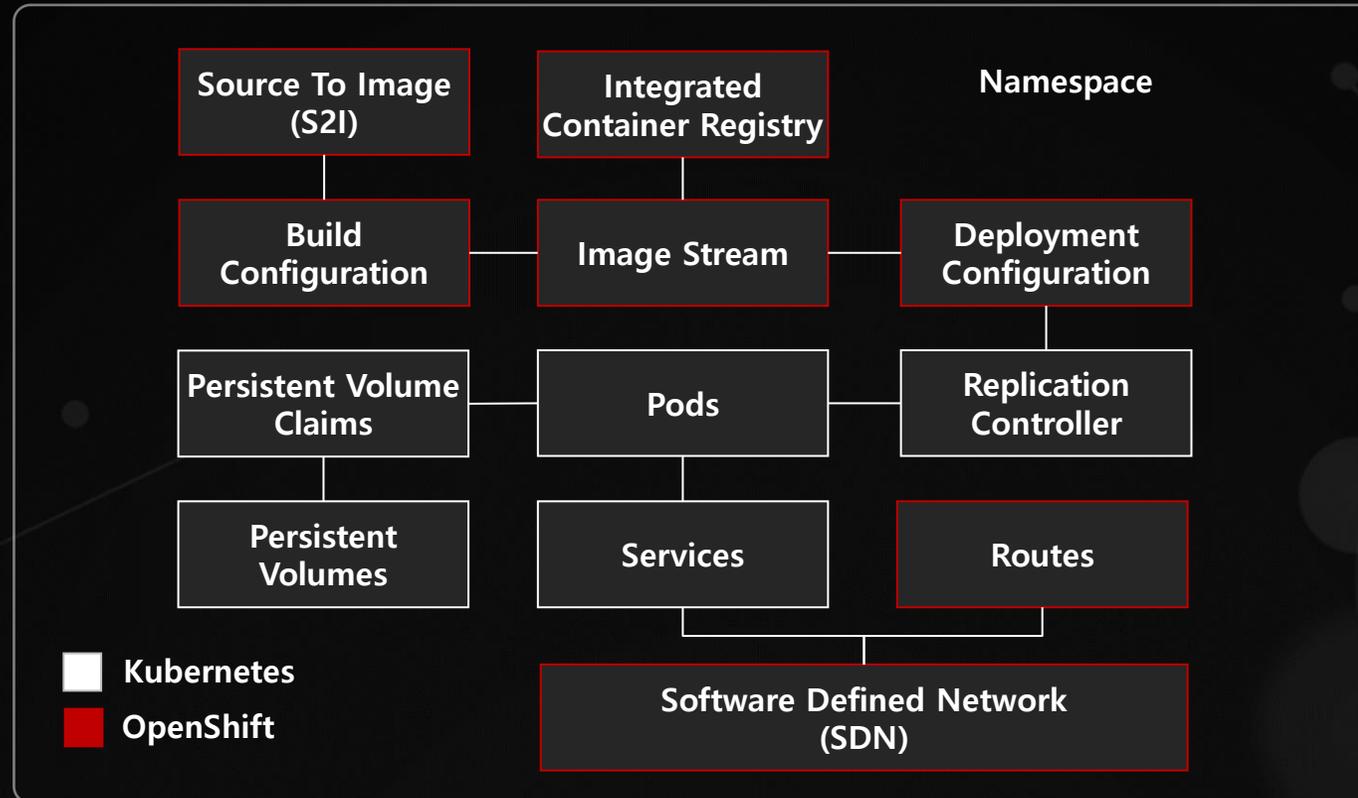
- OpenShift 에서 추가된 새로운 개념이 프로젝트
- 프로젝트를 통해 네임 스페이스에 액세스를 사용자 및 그룹을 기반으로 인증 및 권한 부여 모델을 통해 제어
- OpenShift의 프로젝트는 네임 스페이스 사이의 구분을 두어 사용자 나 애플리케이션이 액세스



<https://medium.com/level-consulting/the-differences-between-kubernetes-and-openshift-ae778059a90e>

# The Differences Between Kubernetes and OpenShift

- OpenShift project (Namespace)

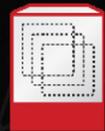


Kubernetes	OpenShift
Ingress Resource ( rules )	Route (rules)
Ingress Controller (NGINX container)	Router ( HAProxy container )

Platform As A Service

# OpenShift 소개

# OpenShift == Enterprise Kubernetes



RED HAT  
ENTERPRISE LINUX  
ATOMIC HOST



Container  
Host



Container  
API



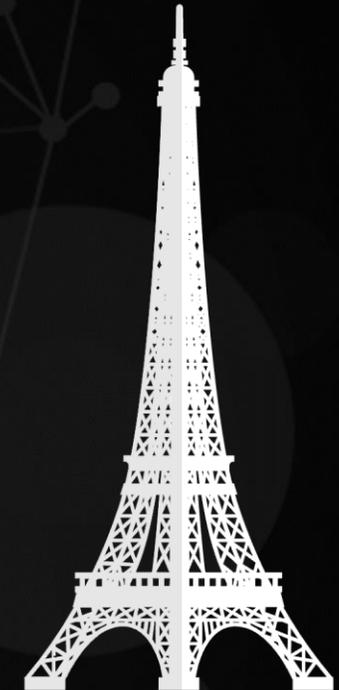
kubernetes

Orchestration



xPaaS

Middleware  
& Services

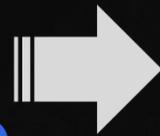


Lifecycle,  
Project Mgt.  
User Experience

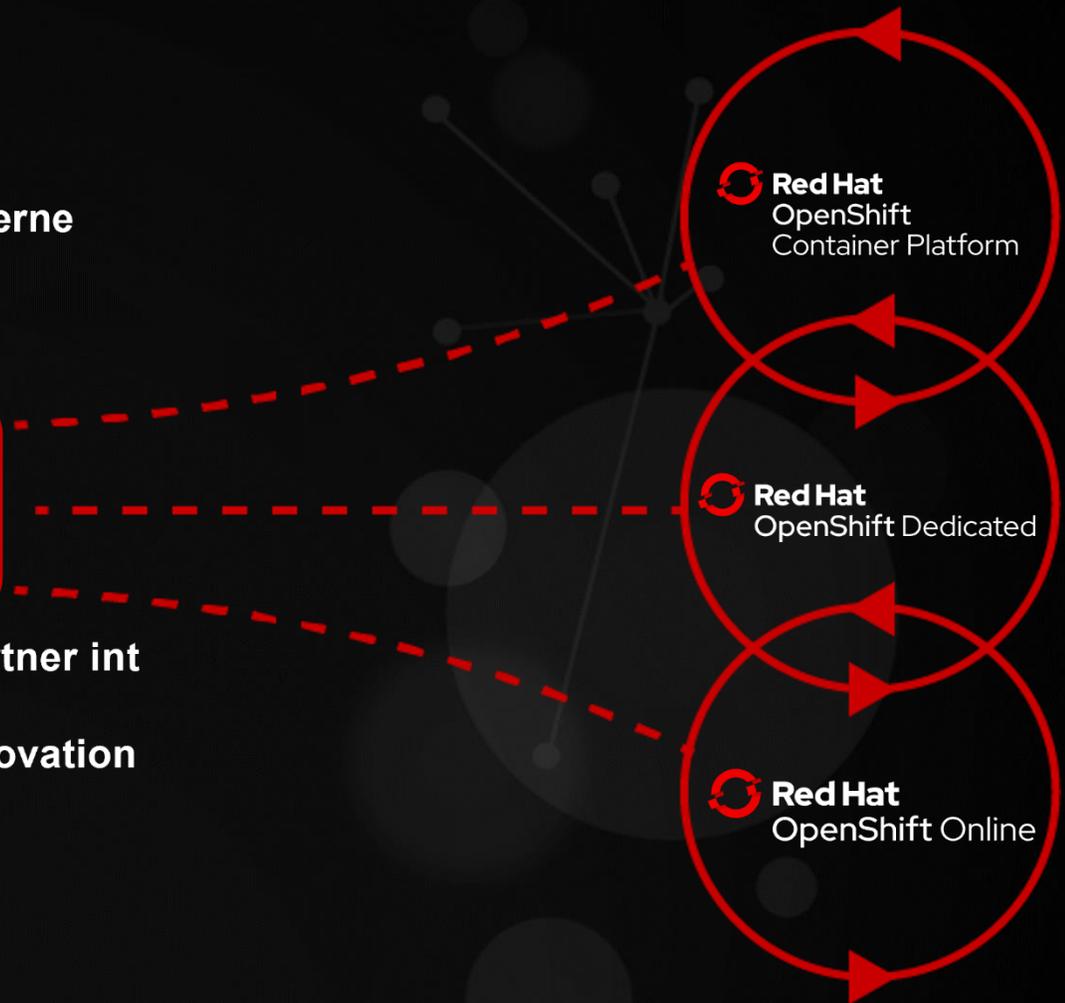
# How Do We Deliver OpenShift?



Community Distribution of Kubernetes 100+ Integrations  
Align time with OSS trunk



Integrate OSS projects Partner integration platform  
No-cost validations for innovation



# OpenShift 가 제공하는 가치



## 신속한 어플리케이션 배치와 DevOps 환경

- 생산성의 높은 시스템 개발 환경
- 민첩한 어플리케이션 운영 환경 제공해
- 비즈니스 기여



## 폭발적인 Openshift 도입 증가

- 14 개의 업종에서 70 이상의 고객사에서 도입
- OpenShift 를 이용한 DevOps 환경 확산



## Opensource 를 기반으로한 기술 혁신

- Red Hat 은 OpenShift 커뮤니티 뿐만이 아니라, 관련한 Docker 나 Kubernetes ,Project Atomic 등 다수의 기술 혁신을 선도



## 엔터프라이즈 PaaS 플랫폼

- 기업에서 안심하고 사용할 수 있는 DevOps플랫폼
- OS 이외의 미들웨어를 제공
- 진정한 의미의 오픈 하이브리드 클라우드 환경

# OPENSIFT IS THE BEST CHOICE FOR KUBERNETES

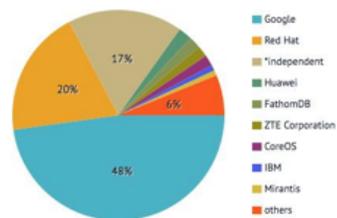
## CUSTOMERS



1000+ customers and the largest amount of reference customers running in production.

Years of experience running OpenShift Online and OpenShift Dedicated services.

## CODE



Red Hat is the leading Kubernetes developer and contributor with Google since day 1.

We make container development easy, reliable, and more secure.

## CLOUD



Strong partnerships with cloud providers, ISVs, CCSPs, (G)SIs.

Extensive container catalog of certified partner images.

## COMPREHENSIVE

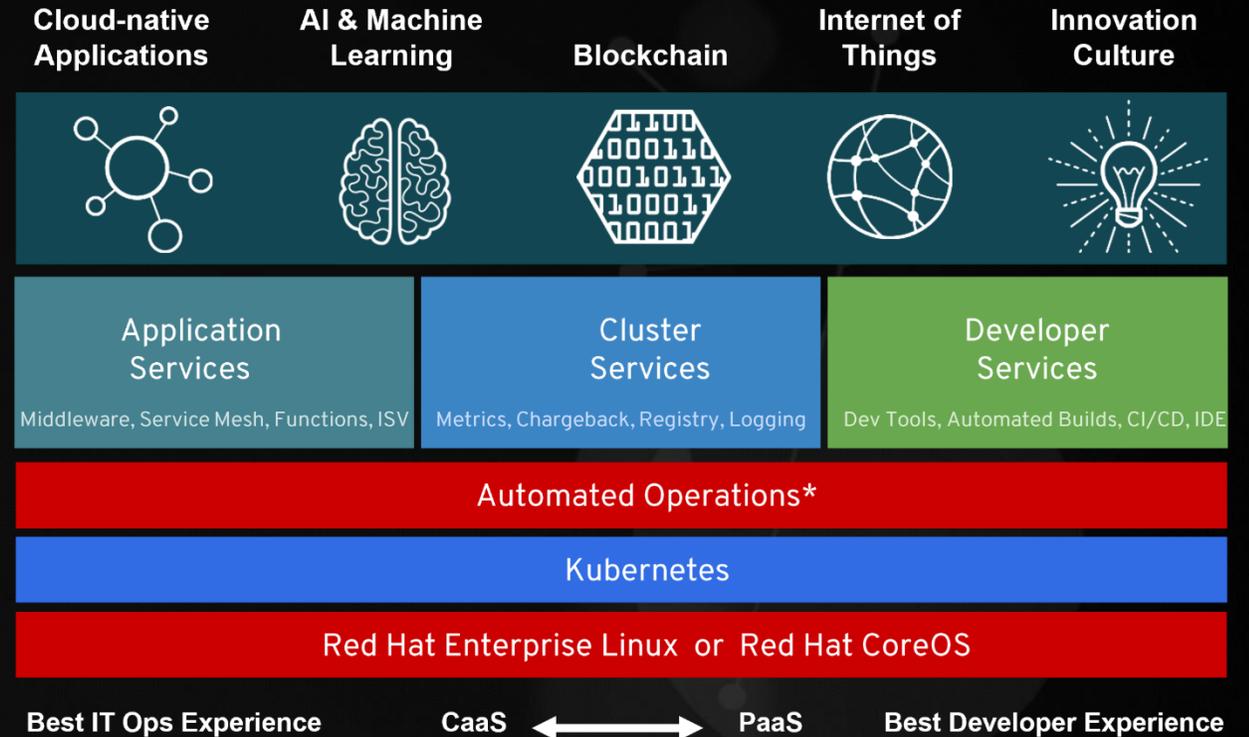


Comprehensive portfolio of container products and services for the enterprise, including developer tools, security, application services, storage, and management.

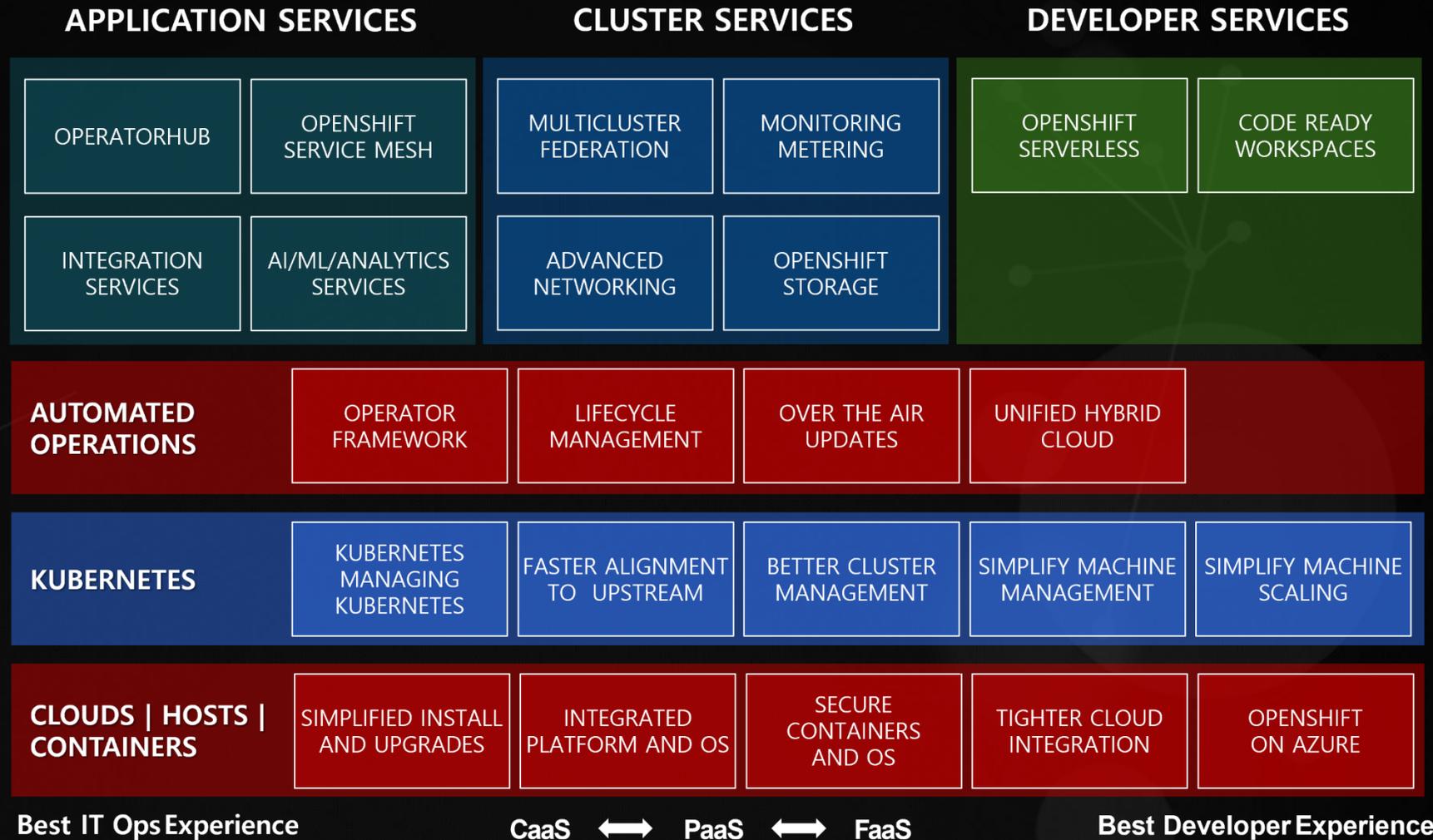
# Enterprise Kubernetes



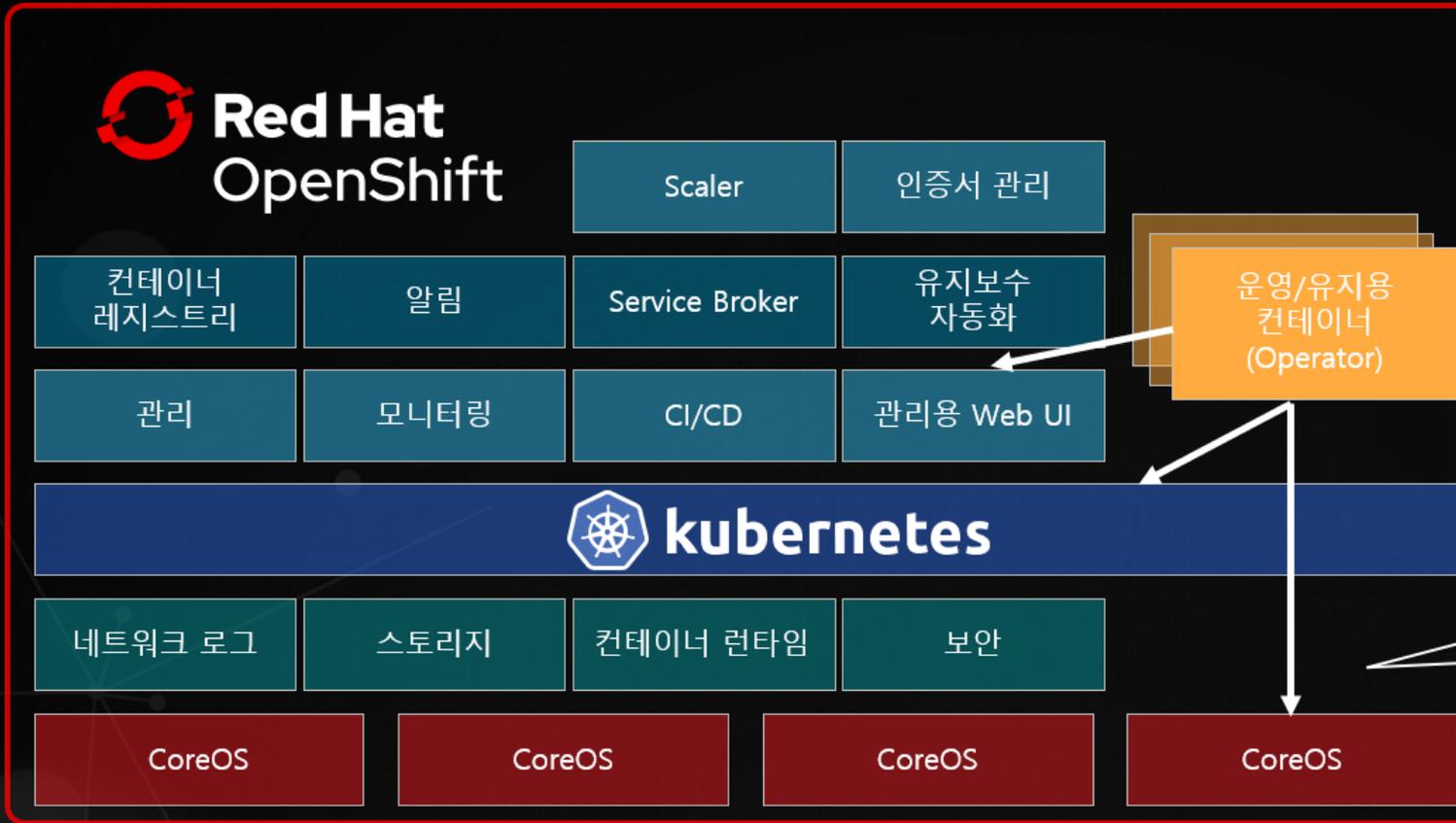
- 신뢰할 수 있는 Enterprise Kubernetes
  - 신뢰할 수 있는 호스트, 콘텐츠, 플랫폼
  - Full-Stacked (전자동) 설치
  - Over the Air Updates & Day 2 management
- 어떤 환경에서도 클라우드를 경험
  - 하이브리드, 멀티 클러스터 관리
  - 오퍼레이터 (Operator) 프레임 워크
  - Operator Hub 와 certified ISVs
- 혁신을 위한 개발자 지원
  - OpenShift Service Mesh (Istio) – 서비스 메쉬
  - OpenShift Serverless (Knative) - Serverless
  - CodeReady (Che) - k8s native 개발 환경



# OpenShift 4 Platform



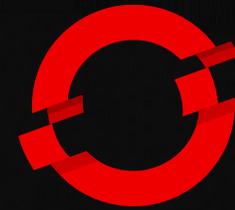
# 컨테이너 플랫폼 구축 및 관리 - OpenShift4



구축 및 운영 관리



- 편하게 **구축** 하는
- 편하게 **유지 보수** 하는
- 편하게 **개발** 하는
- 편하게 **확장** 하는



# Red Hat OpenShift

Platform As A Service

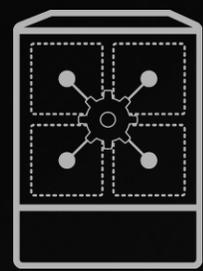


OpenShift 기대효과

## Why OpenShift?

- 엔터프라이즈 급 애플리케이션 지원
  - 기존의 애플리케이션 (상태)에서 클라우드 네이티브 애플리케이션 (무상태, 12 Factor App) 까지
- 업계 표준을 기반
  - CRI-O, Kubernetes, RHEL
- 기업용 미들웨어 서비스
  - JBoss EAP, JWS A-MQ, Fuse, BRMS, BPMS, JDG, Mobile, SSO
- 다양한 운영 환경 지원
  - 물리적 환경, 가상 환경, 프라이빗 클라우드, 퍼블릭 클라우드
- 하이브리드 클라우드 용 애플리케이션 플랫폼
  - 퍼블릭 / 프라이빗 클라우드 간 이동성 보장, 클라우드 프로바이더 간 관리
- Red Hat 을 통한 기술지원
  - 검증된 오픈 소스 리더

# Why customers choose Red Hat OpenShift



**Trusted enterprise  
Kubernetes**



OPENSIFT



**Hybrid/Multi Cloud  
Support**



**Empowering  
developers to innovate**



**Open source innovation**

# Red Hat OpenShift Container Platform

- 컨테이너 형 가상화 기술의 오픈 소스를 기반으로 엔터프라이즈 급 기능 확장 · 품질 향상된 PaaS 기반



■ 즉시 업무 개발에 사용할 수 있는 컨테이너 이미지를 제공  
Java, Node.js, Python, Ruby, Perl, PHP, PostgreSQL,  
MySQL / MariaDB, MongoDB, Redis, Jenkins 등

- 웹 기반의 직관적인 인터페이스를 통해 컨테이너 환경의 관리와 컨테이너 이미지 인출
- 사용자 관리 및 권한 인증 및 인가 멀티 테넌트 환경 제공
- 감시 기능의 강화  
(컨테이너 당 CPU / 메모리 사용률 모니터링 · 수집 등)
- 로그 관리 기능 강화
- 소스 코드에서 컨테이너 이미지를 생성하는 기능
- 개발자를 지원하는 기능 제공  
(개발 프로젝트의 관리 및 CI / CD 도구 제공)

■ 사실상의 표준 오픈 소스 기술의 채택

# OpenShift 기대 효과 (1/2)

No	분류	내용	효과
1	개발 및 운영 환경 구축 셀프 서비스	애플리케이션의 이미지를 템플릿 화하여 인프라 팀의 관여 없이, <b>개발 팀이 환경을 구축하고 폐기 및 재생성 가능</b>	<ul style="list-style-type: none"> <li>• 신속한 개발 및 운영 환경 구축</li> <li>• 운용 팀 업무 부담 경감</li> </ul>
2	배포 자동화	<b>배포 자동화</b> 를 통해 애플리케이션을 환경에 따라 복잡하게 설정하고 스크립트를 작성하는 수고를 줄이고 배포 시간을 단축	<ul style="list-style-type: none"> <li>• 배포 시간 단축</li> <li>• 운용 팀 업무 부담 경감</li> <li>• 효율적인 DevOps 실현</li> </ul>
3	환경 표준화	<b>동일한 애플리케이션 이미지를 개발/스테이징/운영/DR 환경</b> 에서 사용할 수 있어 환경 간의 다른 차이를 제거	<ul style="list-style-type: none"> <li>• 환경 사이의 불일치 제거</li> <li>• 운용 팀 업무 부담 경감</li> </ul>
4	여러 버전 동시 실행	동일한 애플리케이션의 <b>다른 버전을 동시에 운영</b> 하는 것이 가능하여, 기존 버전을 둔 채로 새로운 버전의 개발과 배포가 가능하여 <b>서비스 중단없이 버전의 전환이 가능</b>	<ul style="list-style-type: none"> <li>• 빈번한 Bug Fix 를 위한 잦은 배포 가능</li> <li>• 개발 팀의 지속적인 개발</li> <li>• 무 중단 서비스 업그레이드 가능</li> </ul>
5	마이크로 서비스 화 촉진	컨테이너 단위로 마이크로 서비스를 개발하고 관리함으로써 변화에 유연하게 대응하는 것이 용이	<ul style="list-style-type: none"> <li>• 급격한 부하에도 유연하게 확장</li> <li>• 장애의 영향 범위를 최소화</li> </ul>

# OpenShift 기대 효과 (2/2)

No	분류	내용	효과
6	효율적인 H/W 자원 활용	컨테이너화하여 가상화보다 더 많은 애플리케이션을 동일 HW위에서 움직이는 것이 가능 컨테이너는 애플리케이션이 사용되는 경우에만 자원을 이용	<ul style="list-style-type: none"> <li>동일한 인프라에서 더 많은 처리가 가능</li> <li>인프라 비용 절감</li> </ul>
7	네트워크 관리 작업의 효율화	애플리케이션에 액세스 OpenShift 라우팅 레이어가 수행하는 각 애플리케이션의 네트워크 설정이 불필요	<ul style="list-style-type: none"> <li>운영 팀의 부하 경감</li> </ul>
8	인프라 환경에 독립적	OpenShift 개인/퍼블릭 클라우드 베어 메탈 등 다른 환경에서 작동 가능하며, 인프라 독립적이기 때문에 예산·요구 사항에 맞는 환경을 선택할 수 인프라가 바뀌어도 애플리케이션에 미치는 영향을 최소화할 수	<ul style="list-style-type: none"> <li>운영과 관련된 비용에 따라 인프라의 변경에 유연하게 대응 가능</li> </ul>
9	애플리케이션의 불변성	컨테이너에서 동작하는 애플리케이션 개발/ QA /준비/프로덕션 환경에서 동일하므로 프로덕션 이전 테스트 작업을 간소화 할 수있게 프로덕션 환경에 배포 시간을 단축	<ul style="list-style-type: none"> <li>테스트 공수의 삭감</li> <li>Time to market 단축</li> <li>환경 사이의 차이의 감소</li> </ul>
10	혁신의 촉진	환경의 구축·삭제가 용이 해져 최소의 비용으로 실현 가능하기 때문에 신규 서비스 개발을 트라이 할 용이해진다	<ul style="list-style-type: none"> <li>혁신을 통한 비즈니스 기회 확대</li> <li>"실패"비용을 절감</li> </ul>

## PaaS 구축 목적

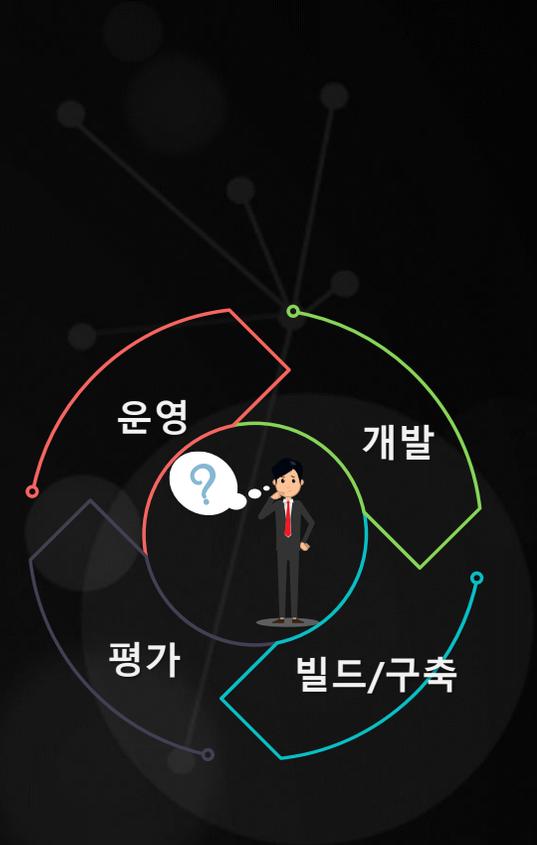
- 고성능 클라우드 시스템을 쉽고 빠르게 구축
  - Private Cloud Platform 으로 검증된 OpenShift 를 제공하여, 안심하고 클라우드 서비스를 이용
  - 어플리케이션 사용자의 증감에 따라 자동으로 스케일링하여 성능과 품질을 향상하고 유지
  - 대규모이면서 짧은 사용자 응답시간을 요구하는 시스템에 높은 사양의 클라우드 서비스를 제공
- 애플리케이션 개발 환경 제공
  - OpenShift 환경에서 컨테이너를 운영하기 위한 템플릿을 손쉽게 작성
  - 어플리케이션을 효율적으로 컨테이너로 가동시키기 위한 노하우를 정리해 개발 비용 절감
- PaaS 기반의 운용
  - 운용에 필요한 백업·감시·로그 수집의 기능도 제공하고 싶다
  - 수고를 들이지 않고 , 안정되어 운용할 수 있도록(듯이) 하고 싶다

## PaaS 구축 목적

- 고성능 클라우드 시스템을 쉽고 빠르게 구축
  - Private Cloud Platform 으로 검증된 OpenShift 를 제공하여, 안심하고 클라우드 서비스를 이용
  - 사용자의 증감에 따라 자동으로 스케일링하여 성능과 품질을 향상하고 유지
  - 대규모이면서 짧은 사용자 응답시간을 요구하는 시스템에 높은 사양의 클라우드 서비스를 제공
- 애플리케이션 개발 환경 제공
  - OpenShift 환경에서 컨테이너를 운영하기 위한 템플릿을 손쉽게 작성
  - 어플리케이션을 효율적으로 컨테이너로 가동시키기 위한 노하우를 정리해 개발 비용 절감
- PaaS 기반의 운용
  - 운용에 필요한 백업·감시·로그 수집의 기능 제공
  - 수고를 들이지 않고 , 안정되어 운용할 수 있도록(듯이) 하고 싶다

## OpenShift를 채용하는데 있어서의 포인트

- Docker를 그대로 움직이는 것과 OpenShift로부터 움직이는 것은 사정이 다르다
  - Docker만으로는 부족한 부분을 커버하고 있다
  - 컨테이너의 빌드와 배포의 구조
  - GUI에 의한 조작
  - 보안이 적용된 컨테이너 운영 환경
- PaaS의 라이프 사이클을 이해하면 장점
  - 템플릿을 만들어 끝은 아닙니다
  - 템플릿이 움직여 끝은 아닙니다
  - 기존 AP를 움직여 끝은 아닙니다
- CI/CD로 서비스를 계속 제공해 변화에 추종하는 것이 OpenShift의 메리트



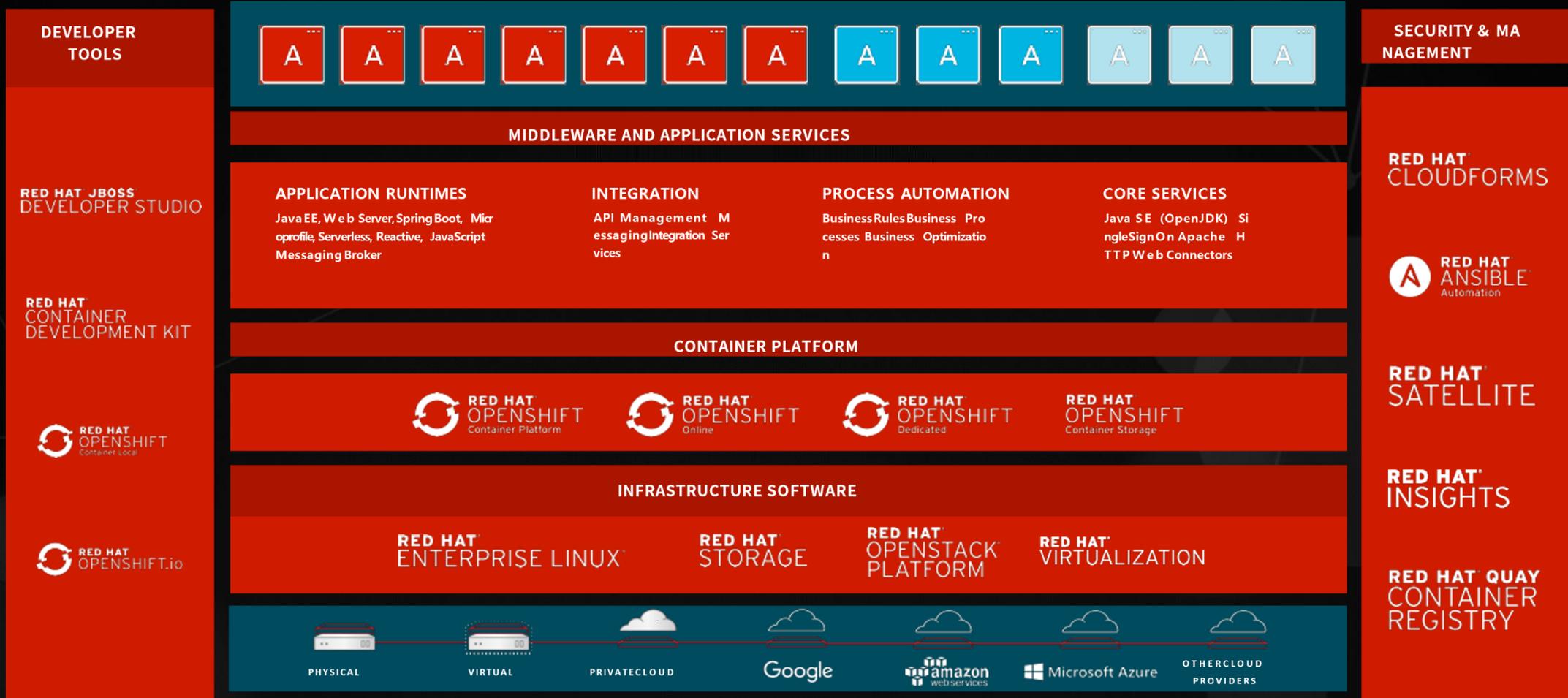
Platform As A Service

# OpenShift Portfolio

# IT 'S ALL THERE!



- 오픈 하이브리드 클라우드를 위한 완벽한 기술 스택



# OpenShift 지원 에코 시스템



- Portfolio / Third Party ISV / Cloud Service with Support

## RED HAT PORTFOLIO

Optimized for Containers

RED HAT  
OPENSIFT  
Application Runtimes

RED HAT JBOSS  
WEB SERVER

RED HAT JBOSS  
ENTERPRISE  
APPLICATION PLATFORM

RED HAT  
DATA GRID

RED HAT  
AMQ

RED HAT  
FUSE

RED HAT  
MOBILE

RED HAT  
ANSIBLE  
Engine

RED HAT QUAY  
CONTAINER  
REGISTRY

RED HAT  
DECISION  
MANAGER

RED HAT  
PROCESS AUTOMATION  
MANAGER

RED HAT 3SCALE  
API MANAGEMENT

RED HAT  
OPENSIFT  
Container Storage

## THIRD-PARTY ISV

Red Hat Container Catalog (200+ certified)

IBM

Microsoft

SAP

New Relic

Couchbase

CRUNCHY  
Enterprise PostgreSQL

Sysdig

f5

VERITAS

ZABBIX

NetApp

Sonatype

Jfrog

dynatrace

nuagenetworks

## CLOUD SERVICES

Open Service Broker

amazon  
web services

Microsoft Azure

Google Cloud



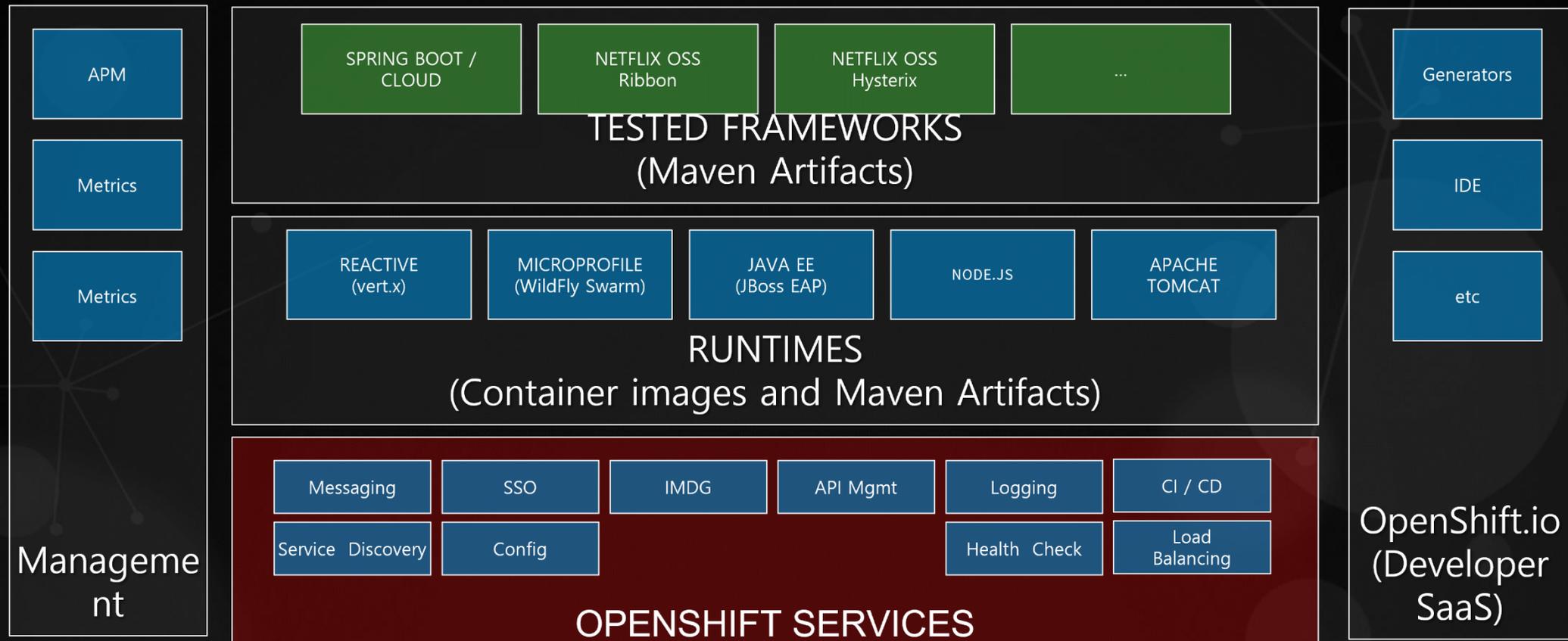
## RED HAT ENTERPRISE LINUX ECOSYSTEM

Hardware, Virtualization, Cloud and Service Provider Certifications

# Red Hat OpenShift Application Runtimes



- Red Hat OpenShift Application Runtimes는 개발자들이 OpenShift에서 새로운 애플리케이션을 개발하고 기존 애플리케이션을 현대화하기 위해 실행 환경과 프레임워크를 선택할 수 있게 합니다.



# OpenShift Commons Participants

- 200사 이상 OpenShift 사용자, 기여자, 파트너, 서비스 제공자, 개발자가 아이디어, 소스 코드, 모범 사례 및 경험을 공유하는 커뮤니티



Source - <https://commons.openshift.org/participants.html>



Red Hat OpenShift

References

# OpenShift 활용 패턴

## 패턴 1 : 컨테이너 플랫폼 (Lift & Shift, Refactor Apps)

- 오렌지라이프 (MIS용 컨테이너 플랫폼 구축 사업)
- 국가정보자원관리원 (자동자원 확장플 사업)
- 삼성전자 (PaaS for MIS)

- 두산정보통신 (그룹사 PaaS 서비스)
- 신한 은행 (글로벌뱅킹 대외계 Open API 시스템 구축)
- 정보통신산업진흥원 (NIPA 클라우드 지원센터)

## 패턴 3 : 하이브리드 클라우드

## 패턴 2: 클라우드 네이티브 애플리케이션

- (그룹사 PaaS 서비스) eBay Korea -
- (All Credit 서비스 Microservices) KCB -

- (차세대 LP 플랫폼 구축 사업) 롯데카드 -
- (국민참여형 도시문제해결 플랫폼-compas) LH공사 -
- (온-나라 시스템 문서 2.0 클라우드) 행정자치부 -

## 패턴 4 : 비즈니스 혁신



# RHOCP CUSTOMER CASE STUDY - 롯데카드



롯데카드

“PaaS 플랫폼을  
이용하여 MLC  
통합 시스템을 구축 및  
DevOps 환경의  
효율적인  
IT 운영 환경 구현”

## CHALLENGE

- MLC 통합 시스템을 운영하기 위한 안정적이고 분산화된 플랫폼 필요
- DevOps 기반의 효율적인 IT 운영 환경 구현 필요

## SOLUTION

- 채널 시스템의 전 영역 기반 플랫폼으로 OpenShift 구축
- Atlassian ALM(Jira, Confluence, Bitbucket, Bamboo)와 연계한 DevOps 환경 구축

## WHY RED HAT

- 안정성 및 보안성이 검증된 OpenShift 선택

## RESULTS

- 채널계 모바일, 웹, 백엔드 시스템의 경량화로 Mass Marketing 가능
- 개발 프로세스 단축 및 Time to Market 실현
- 롯데카드 전 영역의 컨테이너 전환 준비 중

## PRODUCTS & SERVICES

- Red Hat OpenShift Container Platform
- Red Hat Discovery Session
- Red Hat Design Workshop



롯데카드,  
'레드햇 포럼  
서울 2018'  
최고 디지털  
전환상 수상



## CHALLENGE

- 글로벌 대외계 API Gateway 서비스를 위한 PaaS 클라우드 플랫폼 구축 필요
- 신기술에 대한 신한은행 기술 내재화 필요
- MSA 개발 방법론 수용 가능한 플랫폼 도입 필요

## SOLUTION

- 글로벌 대외계 API 서비스를 위한 하이브리드 클라우드 기반의 컨테이너 플랫폼 구축
- 컨테이너 기반의 CI/CD 프로세스 정립

## WHY RED HAT

- 표준 컨테이너 기술 및 오픈소스 기술이 적용된 PaaS 플랫폼을 제공하여 Public/Private Cloud 환경과 가상화와 베어메탈 환경에 걸쳐 하이브리드 클라우드 환경을 구성
- 컨테이너 기반의 API Management 솔루션(3scale) 제공하여 글로벌 핀테크 비즈니스 가속화

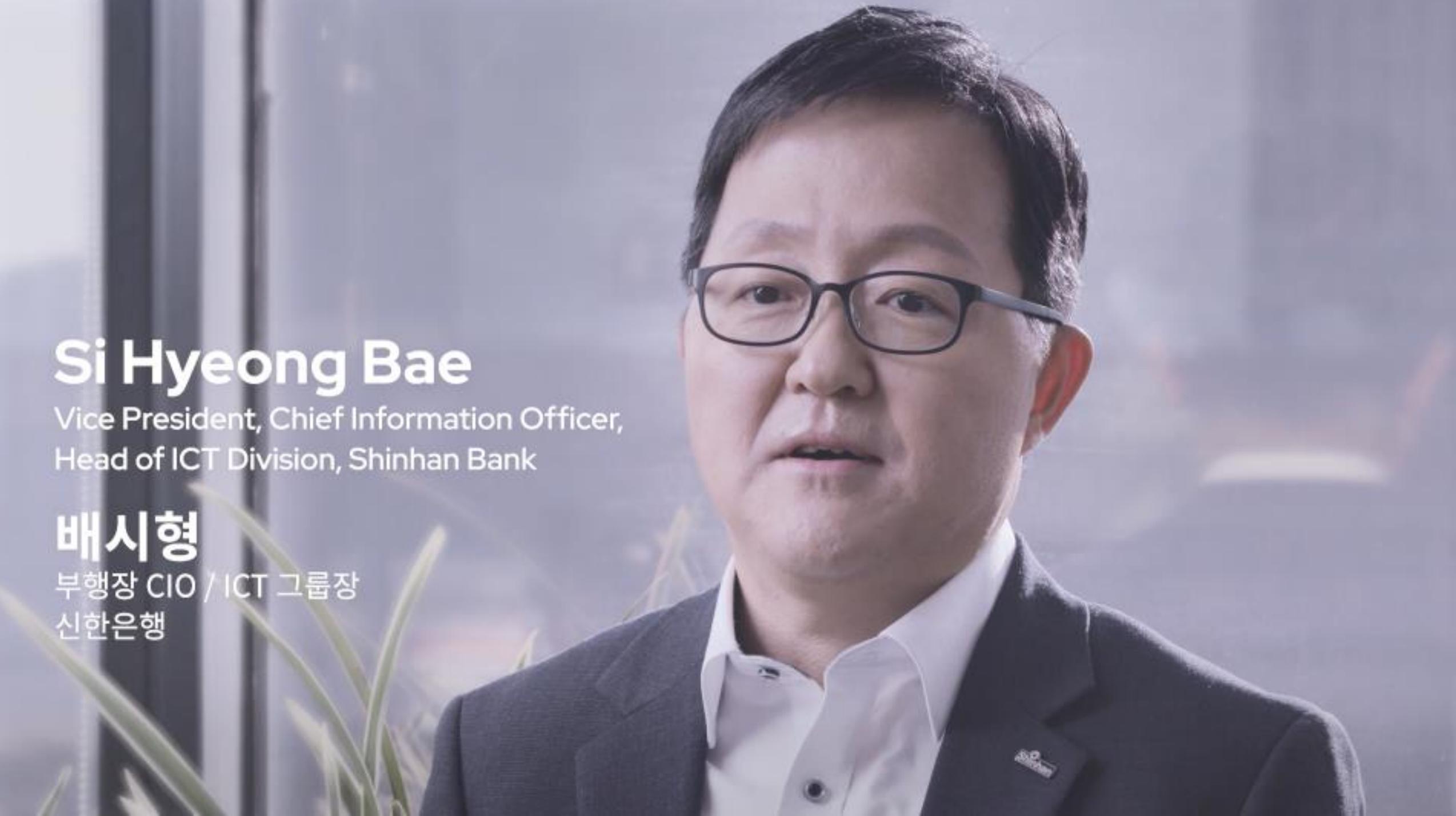
## RESULTS

- 글로벌 Open API 서비스를 위한 PaaS 아키텍처 구축
- 컨테이너 기반의 CI/CD 시스템 운영관리체계 표준화

## PRODUCTS & SERVICES

Red Hat Discovery Session  
Red Hat OpenShift Container Platform

Red Hat 3scale API Management  
Red Hat Fuse

A portrait of Si Hyeong Bae, a middle-aged man with short black hair and glasses, wearing a dark suit jacket over a light-colored button-down shirt. He is looking slightly to the right of the camera. The background is a blurred office interior with a window showing a cityscape. A small green plant is visible in the bottom left corner.

# Si Hyeong Bae

Vice President, Chief Information Officer,  
Head of ICT Division, Shinhan Bank

## 배시형

부행장 CIO / ICT 그룹장  
신한은행

# RHOCP CUSTOMER CASE STUDY - 행정자치부



## 행정안전부

“OpenShift에서 정부부  
처별 개별 운영되던 정부  
표준 그룹웨어를 SaaS  
on PaaS 형태로 구축”

### CHALLENGE

온-나라 애플리케이션은 2005년 개발된 정부 부처와 지방자치단체 공무원이 문서 결재 등을 위해 사용하는 정부 업무시스템으로, 부처별, 개별적으로 유지보수 및 추가 개발하여 소스 버전 관리 및 통합 어려움이 있음

### SOLUTION

- 공개 오픈소스 기반 WAS/DB 전환, 표준 컨테이너 및 PaaS 플랫폼 구축
- On-Nara Application을 SaaS 형태의 클라우드 공통 서비스로 구축

### WHY RED HAT

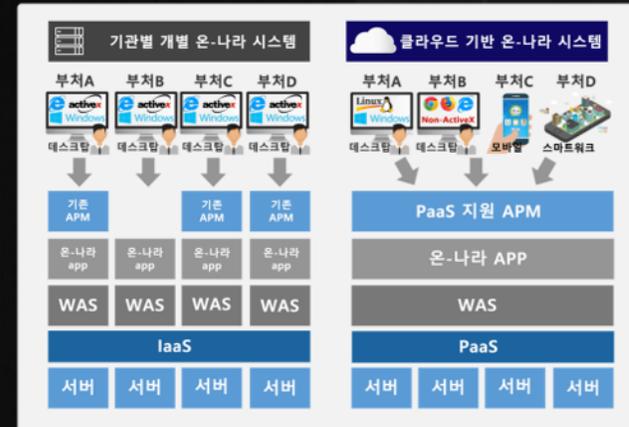
- 표준 컨테이너 기술이 적용된 PaaS 및 오토 스케일링 구현
- 표준 오픈소스 기반의 WAS 제공

### RESULTS

- PaaS 플랫폼 기반의 클라우드 공통 서비스 환경 구축
- 오래된 기술 및 벤더 종속 기술을 탈피하여 최신 표준 기술 적용 (표준 컨테이너, 오픈소스, PaaS)

### PRODUCTS & SERVICES

- Red Hat Virtualization
- Red Hat OpenShift Container Platform





제품 / 서비스에 관한 문의

- 콜 센터 : 02-469-5426 ( 휴대폰 : 010-2243-3394 )
- 전자 메일 : [sales@openmaru.com](mailto:sales@openmaru.com)