

# Open, Hybrid, Multi Cloud Container Platform Red Hat OpenShift

전체

SBS Mtv 2014.05.13

# Agenda

1. Digital Transformation
2. Docker, Kubernetes, OpenShift
3. DevOps, CI/CD, MSA
4. Why Red Hat OpenShift
5. Where & How to start

# Digital Transformation을 어떻게 가능하게 할 것인가?

아래 세가지 영역의 발전이 필요..



## Applications

새로운 방식의 개발,  
딜리버리, 통합



## Platform

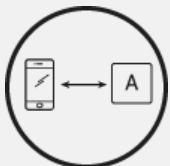
기존 플랫폼 현대화 및  
클라우드 기반 인프라 구축



## Process

IT와 비즈니스의 보다  
민첩한 프로세스

# Containers Transform



Monolith



N-Tier



Microservices

## Applications



Datacenter

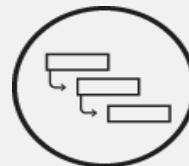


Hosted

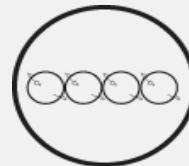


Hybrid

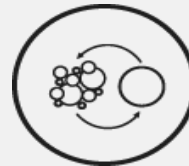
## Infrastructures



Waterfall



Agile



DevOps

## Processes

# Red Hat OpenShift =

# Docker + Kubernetes Enterprise 컨테이너 플랫폼

Docker & Kubernetes 만으로 이룰 수 없는 가치를 제공



docker



Tools



Process



**RED HAT®**  
**OPENS SHIFT**  
Container Platform

# PaaS는 애플리케이션 딜리버리를 가속화

## Craftwork

### Physical

#### How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
- 12. Code**
13. Test
14. Buy and configure Prod servers
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...
19. Deploy new servers
20. Etc.

### Virtualized

#### How to Build an App:

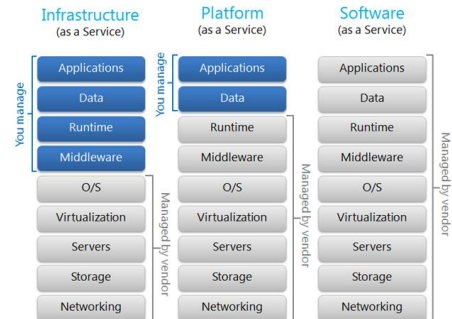
1. Virtualized
2. How to Build an App:
3. Have Idea
4. Get Budget
5. Submit VM Request request
6. Wait
7. Deploy framework/appserver
8. Deploy testing tools
- 9. Code**
10. Test
11. Configure Prod VMs
12. Push to Prod
13. Launch
14. Request VMs to meet demand
15. Wait
16. Deploy app to new VMs
17. Etc.

## Assembly

### With PaaS

#### How to Build an App:

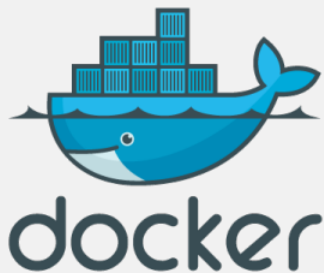
1. Have Idea
2. Get Budget
3. Code
4. Test
5. Launch
6. Automatically Scale



# Docker, Container, Kubernetes



# Docker, Container



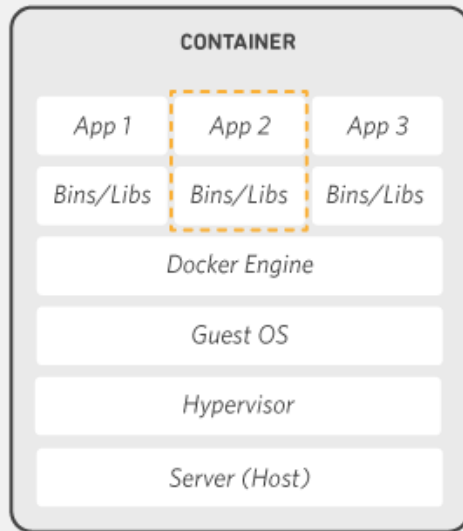
컨테이너 = 프로세스를 격리하여 가상환경처럼 사용

## 역사

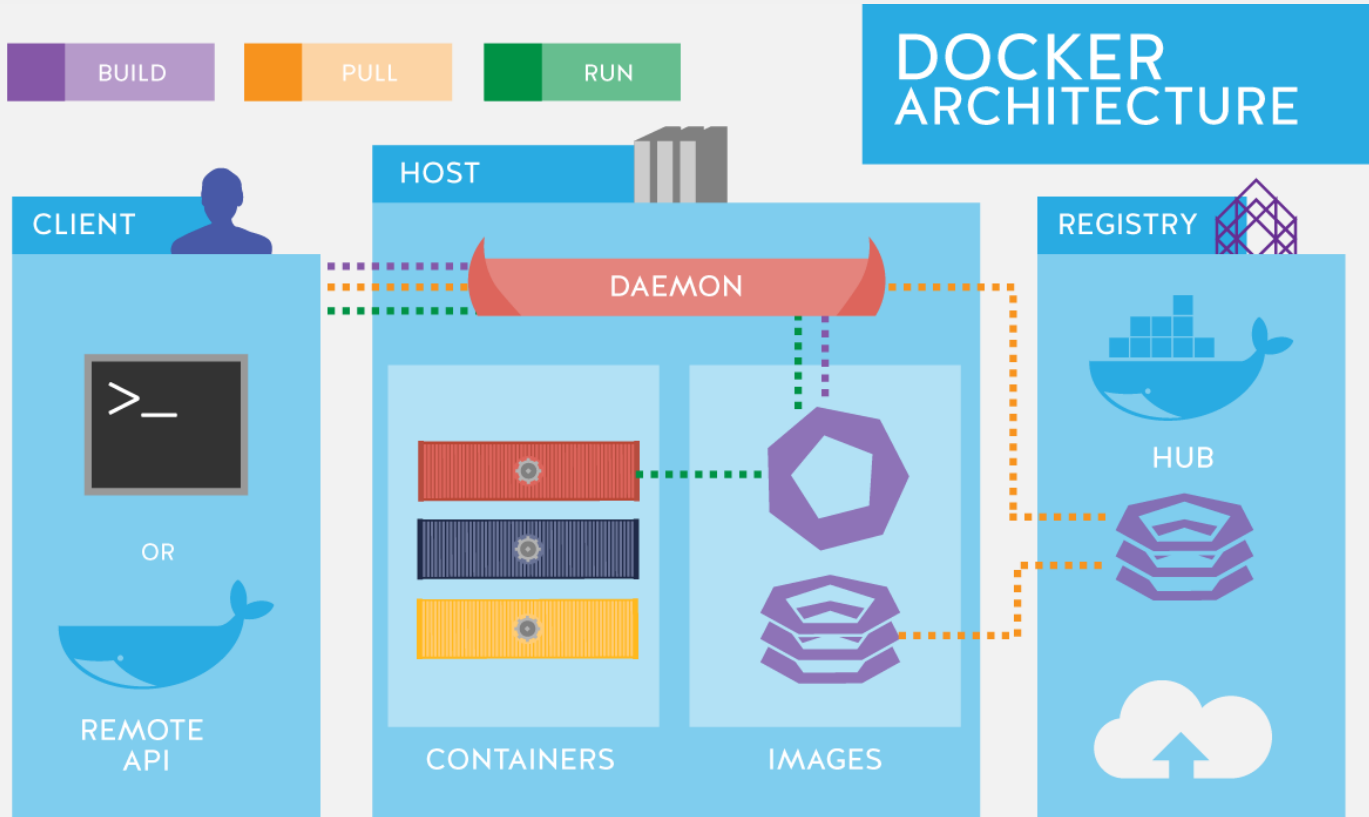
- 2013 PyCon, 솔로몬 하익스, “The future of Linux Containers” 발표 (<https://www.youtube.com/watch?v=wW9CAH9nSLs>)

## 정의

- 컨테이너 기반의 오픈소스 가상화 플랫폼
- “컨테이너”라는 표준화된 유닛으로 소프트웨어를 패키징.
- “컨테이너”에는 라이브러리, 시스템 도구, 코드, 런타임 등 소프트웨어를 실행하는 데 필요한 모든 것이 포함됨
- 애플리케이션을 신속하게 구축, 테스트 및 배포할 수 있는 소프트웨어 플랫폼

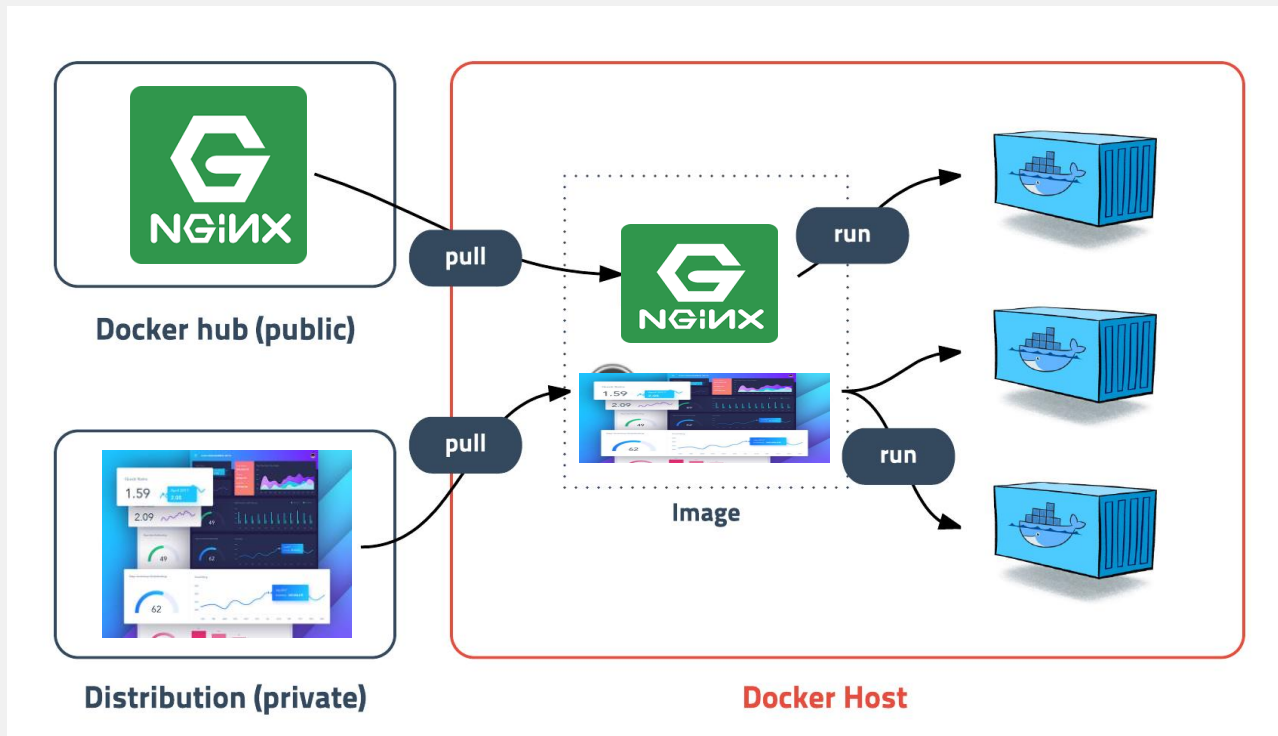
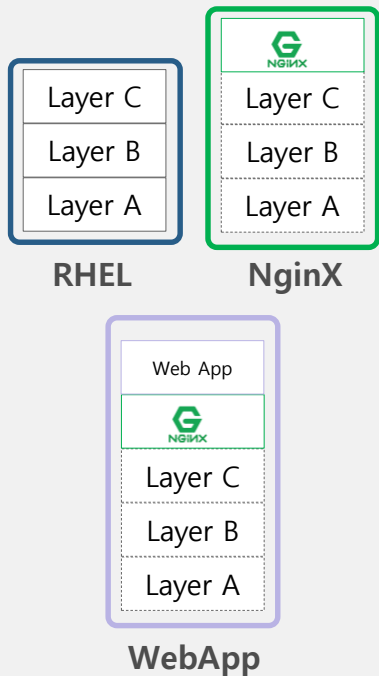


# Docker, Container



NORDICAPIS.COM

# Docker, Container



## Docker Image

컨테이너 실행에 필요한 파일과 설정 값 등을 포함하고 있는 것

## Docker Registry

도커 이미지 저장소

## Docker

컨테이너를 구축, 테스트 및 배포할 수 있는 플랫폼

## (Docker)Container

Docker Image가 Run된 프로세스

# Quick Docker Concepts



## Docker Image

- 컨테이너의 베이스가 되는 것. 프로세스를 실행할 때 필요한 파일들의 집합체, 즉, OS 패키지, 애플리케이션 서버 등 애플리케이션을 실행될 때 필요한 파일들의 집합임. Object Oriented 용어로는 Class로 비교될 수 있음

## (Docker) Container

- 이미지의 인스턴스로 컨테이너로 불림. 하나의 Node/Server에 동일한 이미지로 여러 개의 컨테이너들이 실행될 수 있음. Object Oriented 용어로 Instance와 비교될 수 있음

## Dockerfile

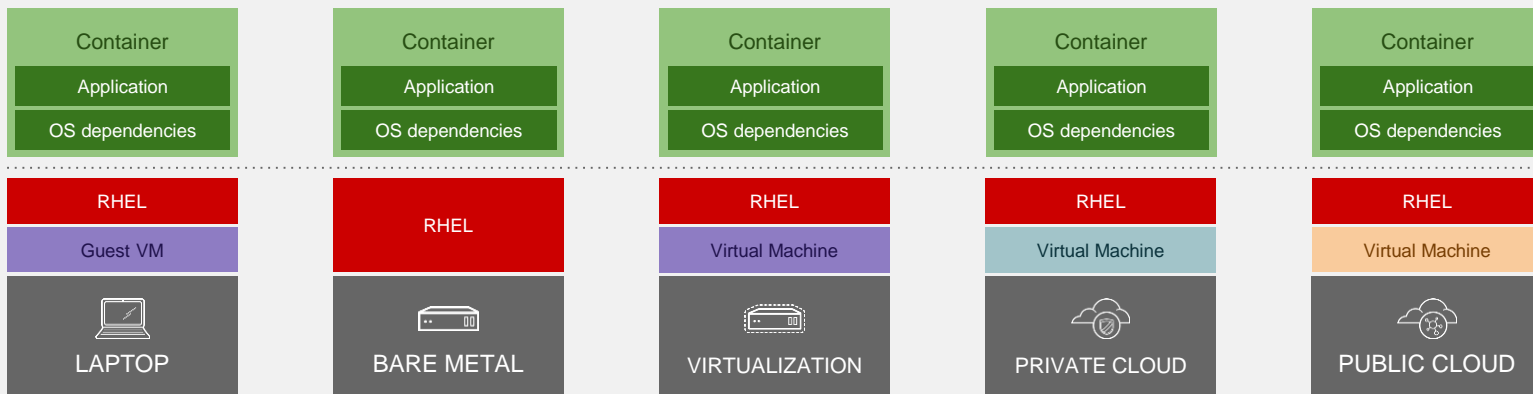
- 도커 이미지를 빌드하기 위해 수작업으로 실행시켜야 하는 커맨드들로 구성된 텍스트 문서. 도커가 도커파일로 부터 명령을 읽어 자동으로 이미지를 빌드함

## Docker Registry

- 이미지들의 저장소이며 관리 서비스. 예를들어 GitHub

# Container – 애플리케이션 디플로이 측면의 발전

- 마이크로서비스뿐만 아니라 전통적인 방식의 애플리케이션의 효율성 및 자동화 가능
- 개발부터 운영시스템까지 보다 빠르고 일관성있는 디플로이
- 베어메탈, 가상화, 프라이빗/ 퍼블릭 클라우드 모든 인프라에 애플리케이션 이식 가능



# Kubernetes (k8s)



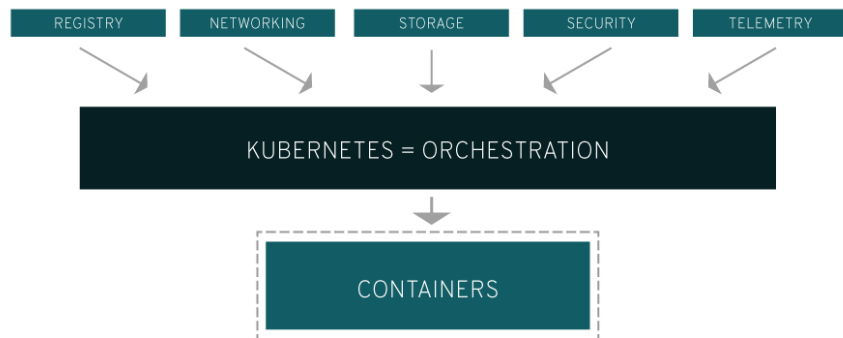
쿠버네티스 = 구글이 만든 도커 컨테이너 오케스트레이션 플랫폼

## 역사

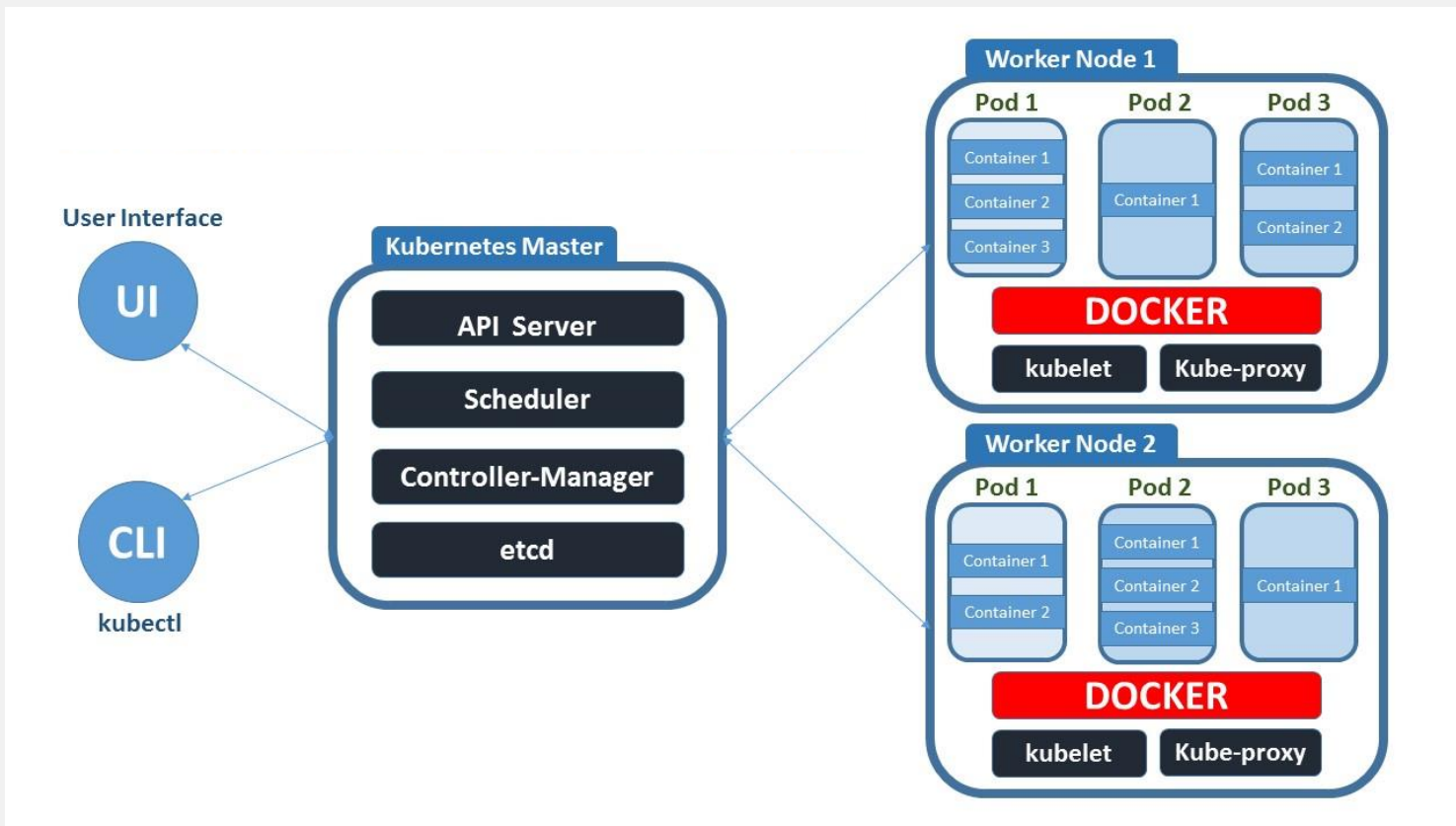
- 2014년 구글의 내부 컨테이너 플랫폼 (Borg)를 오픈소스화 한 것
- 구글의 십 수년간의 대규모 운영 시스템 워크로드 관리 경험의 산물
- Go 언어로 구현됨

## 정의

- 컨테이너 애플리케이션 오케스트레이션, 스케줄링, 디플로이먼트, 확장 자동화
- 구글은 20억+/week 컨테이너 생성, 배포 관리 중



# Kubernetes Architecture



source: [www.learnitguide.net](http://www.learnitguide.net)

# Quick Kubernetes Concepts



## Kubernetes Master

- 쿠버네티스 노드를 관리하는 서버. kube-apiserver, kube-controller-manager, kube-scheduler and etcd 가 실행됨.

## Cluster

- 쿠버네티스가 애플리케이션을 구동 관리하는 물리 서버 또는 VM 의 집합

## Node

- 쿠버네티스가 구동되는 물리서버 또는 VM 단위. Kubelet, proxy, docker 가 구동되고, 노드 내에서 Pod 단위로 스케줄링함

## Pod

- 컨테이너와 볼륨의 그룹 구동 단위. 쿠버네티스가 관리하는 최소 단위. 스케일링, Replication 단위

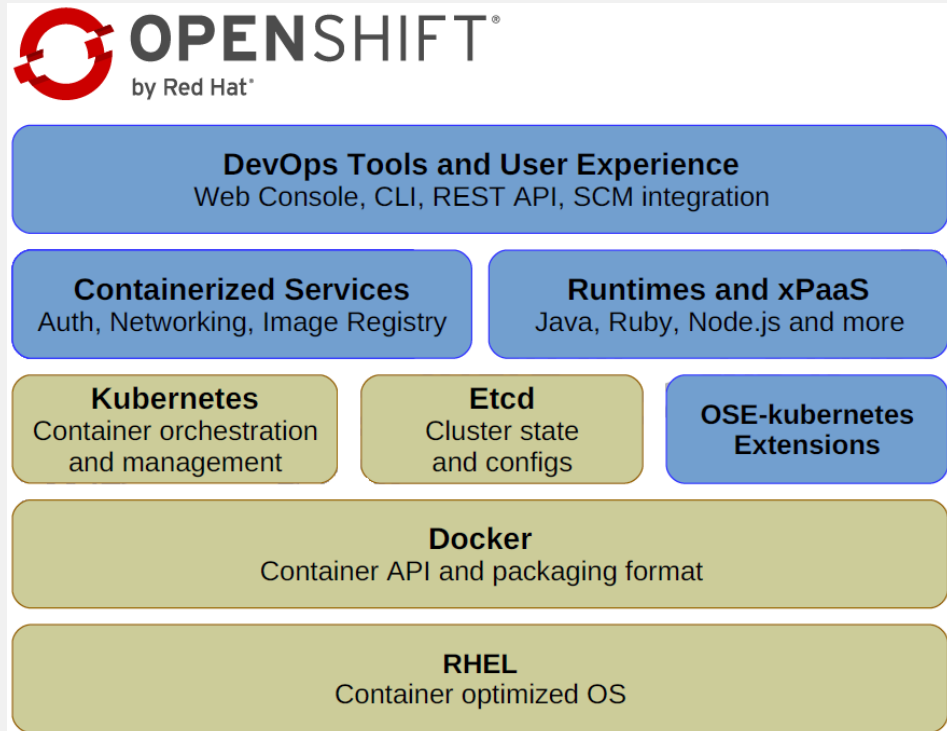
## Service

- Pod의 집합체로 하나의 IP address나 DNS name으로 접근하는 단위



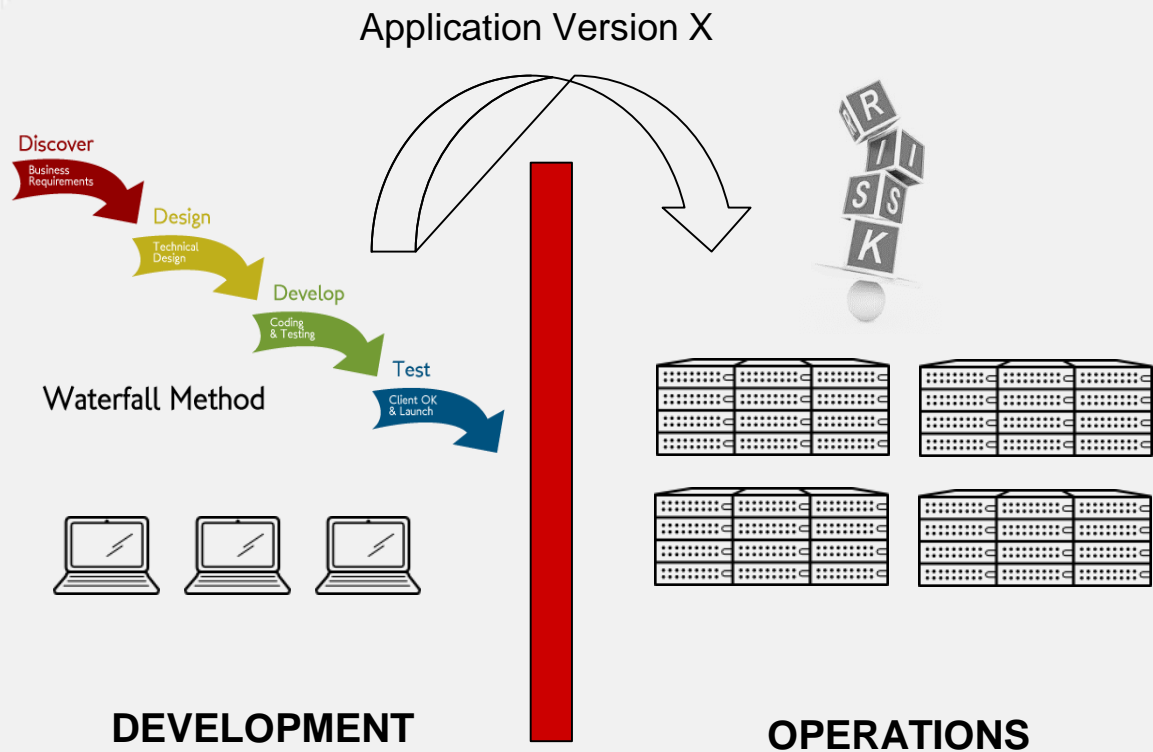
# OpenShift = Enterprise Kubernetes

- OpenShift = Enterprise Kubernetes
- Red Hat은
  - #2 Docker Contributor
  - #2 Kubernetes Contributor
- Docker + Kubernetes +  
Red Hat Application Runtime +  
Open source runtime +  
CI/CD pipeline +  
Security +  
Web Console 등 관리의 편의성



# Microservice, CI/CD, DevOps

# 현재 기업의 IT: Waterfall and Silos



비즈니스

IT의 기민성 요구



Dev

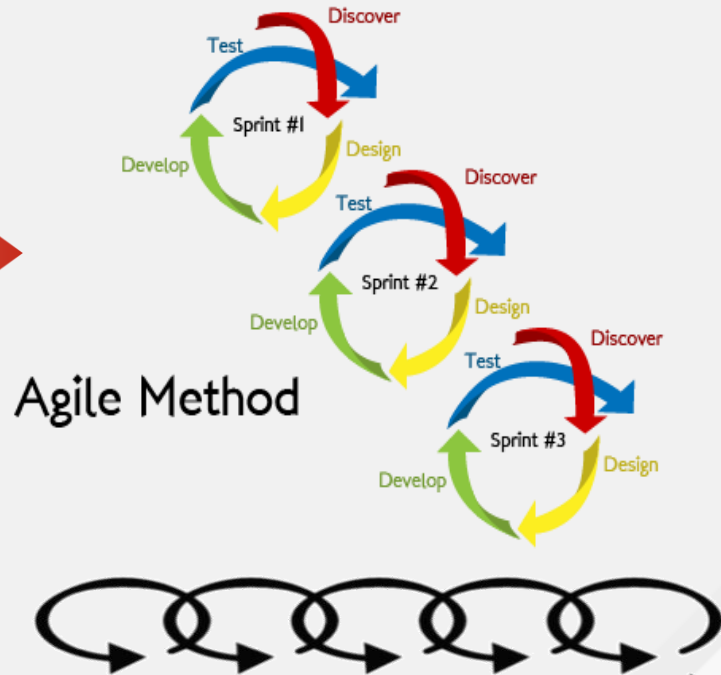
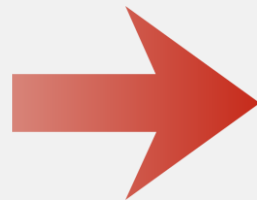
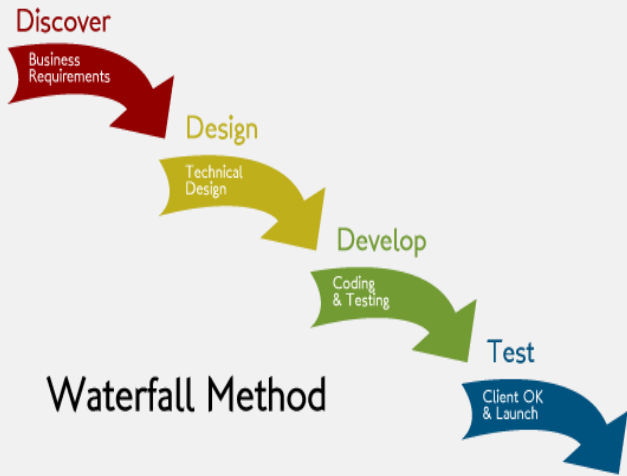
용도에 맞는 기술  
최신 기술, 툴



Ops

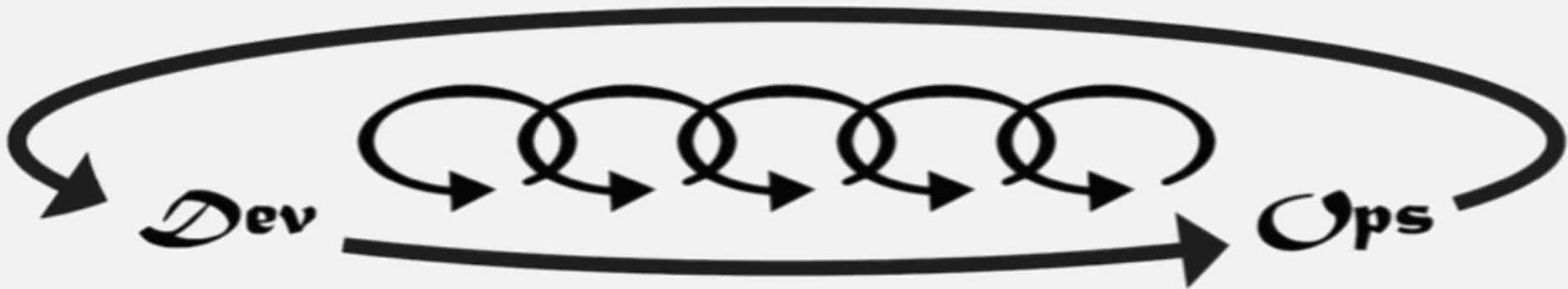
Control  
Risk 최소화

# AGILE SOFTWARE DEVELOPMENT





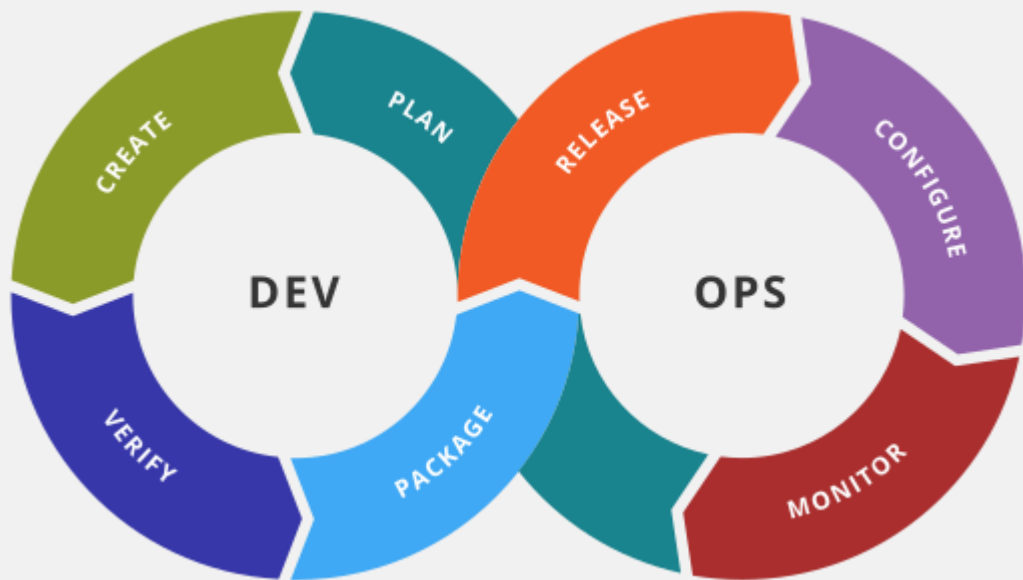
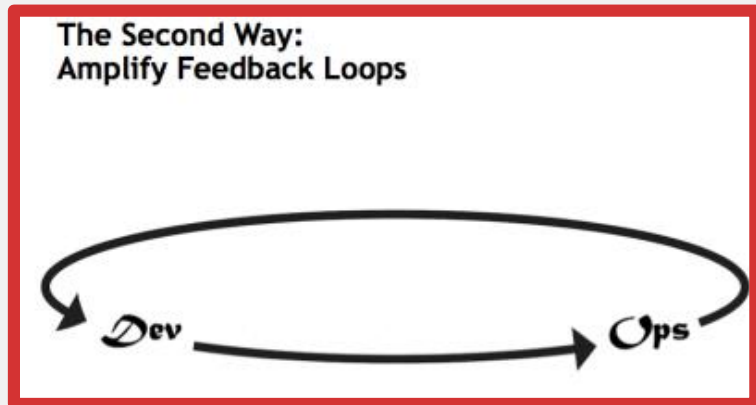
# THE DEVOPS MOVEMENT



## Key concepts:

- 작은 단위로 변경하여 리스크를 줄임
- 빠른 실패를 경험하고 회복 vs. 절대 실패하면 안된다
- 문화의 변화: 실패를 수용

# WHAT ENABLES DEVOPS?



- ✓ **C**ONTINUOUS **I**NTTEGRATION
- ✓ **C**ONTINUOUS **D**ELIVERY

CI/CD: 통합, 테스트, 디플로이 자동화

# WHY RED HAT OPENSIFT



# KUBERNETES **DONE RIGHT** IS HARD

## INSTALL

- Templating
- Validation
- OS Setup

## DEPLOY

- Identity & Security Access
- App Monitoring & Alerts
- Storage & Persistence
- Egress, Ingress & Integration
- Host Container Images
- Build/Deploy Methodology

## HARDEN

- Platform Monitoring & Alerts
- Metering & Chargeback
- Platform Security Hardening
- Image Hardening
- Security Certifications
- Network Policy
- Disaster Recovery
- Resource Segmentation

## OPERATE

- OS Upgrade & Patch
- Platform Upgrade & Patch
- Image Upgrade & Patch
- App Upgrade & Patch
- Security Patches
- Continuous Security Scanning
- Multi-environment Rollout
- Enterprise Container Registry
- Cluster & App Elasticity
- Monitor, Alert, Remediate
- Log Aggregation

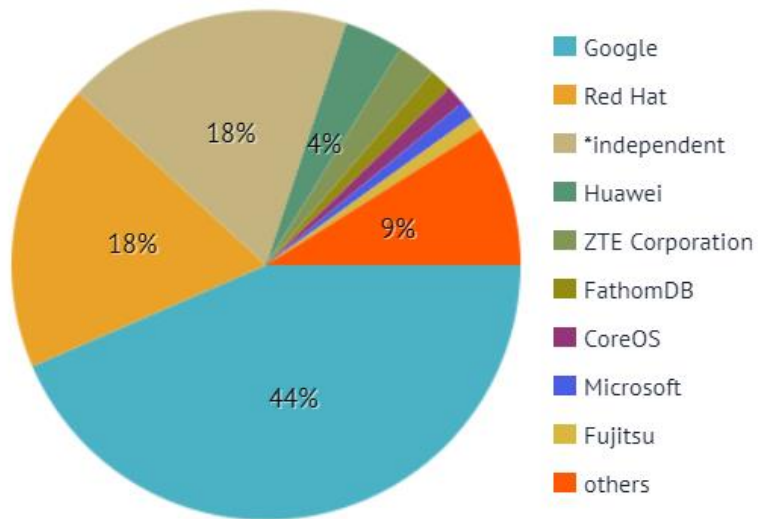
 **75%**

of enterprise users identify complexity of implementation and operations as the top blocker to adoption

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

# Kubernetes 프로젝트 기여도

Contribution by companies

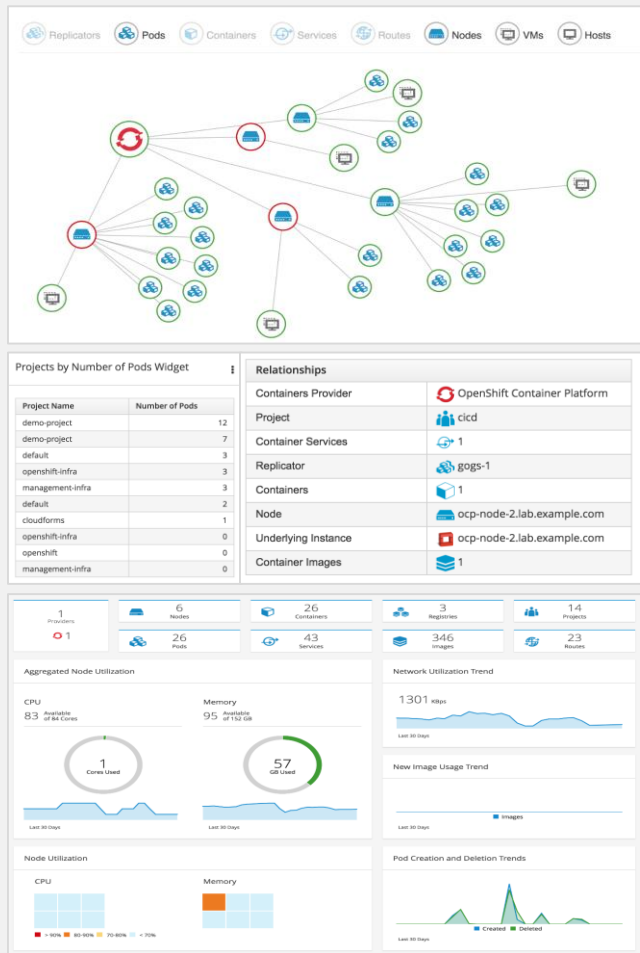


#	Company	Commits
1	Google	17634
2	Red Hat	7423
	*independent	7419
3	Huawei	1521
4	ZTE Corporation	1003
5	FathomDB	585
6	CoreOS	504
7	Microsoft	425
8	Fujitsu	405
9	IBM	378

소스: [http://stackalytics.com/?project\\_type=kubernetes-group&metric=commits&release=all&module=kubernetes](http://stackalytics.com/?project_type=kubernetes-group&metric=commits&release=all&module=kubernetes)

# Enterprise 컨테이너 환경: 운영 효율성

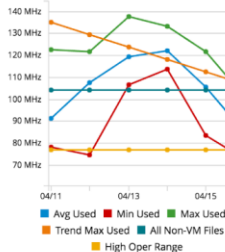
- 인프라 운영 상황 시각화
- 디플로이 노드 정보
- 컨테이너가 디플로이된 노드 정보
- 컨테이너와 연결된 호스트, 스토리지, 다른 컴포넌트
- 문제에 대한 트러블슈팅
- 공통 태스크 자동화



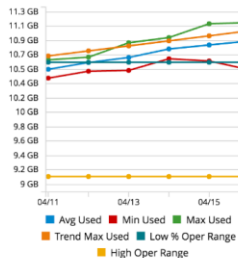
# Enterprise 컨테이너 환경: SERVICE HEALTH

- VM/Storage/Network/컨테이너 리소스 사용 현황 및 트렌드
- Performance threshold에 따른 alert
- Configuration auto detect

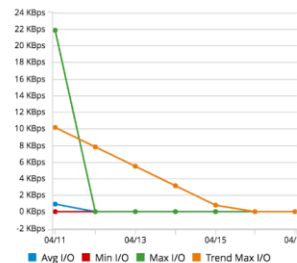
CPU (Mhz)



Memory (MB)



Network I/O (KBps)



Normal Operating Ranges (up to 30 days' data)

	Max	High	Average	Low
CPU	745.90 MHz	705.74 MHz	663.99 MHz	622.23 MHz
CPU Usage	100.00%	15.36%	14.10%	12.84%
Memory	7.7 GB	7.57 GB	7.37 GB	7.18 GB
Memory Usage	65.00%	63.46%	61.78%	60.11%

Right-Sizing (Conservative - derived from Absolute Maximum)

	Current	Recommended	% Savings	Savings
Processors	4	5	-25.0%	-1
Memory	12288 MB	7988 MB	35.0%	4300 MB

Right-Sizing (Moderate - derived from High NORM)

	Current	Recommended	% Savings	Savings
Processors	4	1	75.0%	3
Memory	12288 MB	7800 MB	36.5%	4488 MB

Right-Sizing (Aggressive - derived from Average NORM)

	Current	Recommended	% Savings	Savings
Processors	4	1	75.0%	3
Memory	12288 MB	7596 MB	38.2%	4692 MB

Time Stamp	Type	Name	Event Type	Severity	Message
01/02/18 07:23:10 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	1	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 765.6 GB (100% of Memory Max Total)
11/01/17 06:18:52 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	1	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 689 GB (90% of Memory Max Total)
07/31/17 04:42:25 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	2	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 574.2 GB (75% of Memory Max Total)
02/26/17 02:01:39 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	3	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 382.8 GB (50% of Memory Max Total)

# Enterprise 컨테이너 환경: SECURITY & COMPLIANCE

- 신뢰하지 않는 레지스트리의 베이스 이미지를 사용하는 컨테이너 감지하여 표시

Compliance	
Status	<span style="color: red;">✘</span> Non-Compliant as of 5 Days Ago
History	<span style="color: blue;">↺</span> Available

Configuration	
Packages	528
OpenSCAP Results	431
OpenSCAP HTML	Available
Last scan	Tue, 28 Mar 2017 11:05:54 +0000

OpenSCAP Failed Rules Summary	
Medium	1
High	3

Name	Result	Severity
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20170386	Fail	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20170372	Fail	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20170294	Fail	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20170286	Fail	Medium
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20140685	Pass	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20140686	Pass	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20140679	Pass	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20140703	Pass	Medium
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20140684	Pass	High
xccdf_com.redhat.rhsa_rule_oval.com.redhat.rhsa-def-20140704	Pass	Medium

## Compliance and Scoring

The target system did not satisfy the conditions of 4 rules! Please review rule results and consider applying remediation.

### Rule results

427 passed

### Severity of failed rules

1 medium 3 high

### Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	99.071922	100.000000	99.07%

### Rule Overview

pass  fail  notchecked  notselected  notapplicable  
 fixed  error  unknown  
 informational

Search through XCCDF rules

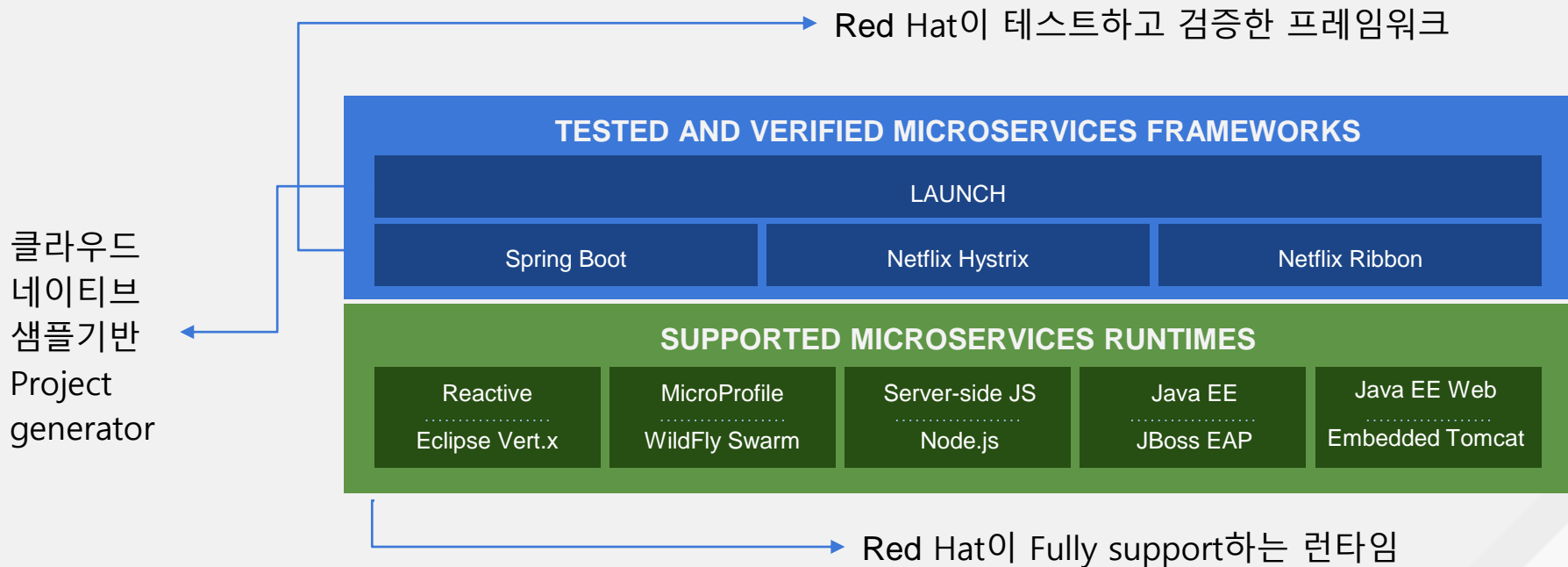
Group rules by: Default

Title	Severity	Result
Automatically generated XCCDF from OVAL file: com.redhat.rhsa-RHEL7.xml <span>fix fail</span>		
RHSA-2017:0286: openssl security update (Moderate)	medium	fail
RHSA-2017:0294: kernel security update (Important)	high	fail
RHSA-2017:0372: kernel-aarch64 security and bug fix update (Important)	high	fail
RHSA-2017:0386: kernel security, bug fix, and enhancement update (Important)	high	fail





# RED HAT® OPENSSHIFT Application Runtimes



# TRUE POLYGLOT PLATFORM

- 오픈스위프트에 탑재된 Certified Docker Images

LANGUAGES	Java	NodeJS	Python	PHP	Perl	Ruby	.NET Core	Third-party Language Runtimes	
DATABASES	MySQL	PostgreSQL	MongoDB	Redis	<p><b>...and virtually any docker image out there!</b></p>			Third-party Databases	CrunchyData GitLab
WEB SERVERS	Apache HTTP Server	nginx	Varnish	Phusion Passenger				Tomcat	Third-party App Runtimes
MIDDLEWARE	Spring Boot	Wildfly Swarm	Vert.x	JBoss Web Server	JBoss EAP	JBoss A-MQ	JBoss Fuse	Third-party Middleware	EnterpriseDB NuoDB
	3SCALE API mgmt	JBoss BRMS	JBoss BPMS	JBoss Data Virt	JBoss Data Grid	RH Mobile	RH SSO	Third-party Middleware	Fujitsu and many more



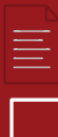
# Red Hat Support

“ We have found the support from Red Hat to be exemplary. Whenever we need anything from them, they have given it... Red Hat is now our backbone. Our business cannot run if Red Hat is not there. ”

ASHISHKUMAR CHAUHAN  
CEO, BSE

[redhat.com/en/resources/bse-case-study](https://redhat.com/en/resources/bse-case-study)

The Red Hat Customer Portal was recognized as  
**ONE OF THE BEST SUPPORT WEBSITES  
FOR SEVEN CONSECUTIVE YEARS**  
by the Association for Support Professionals.\*



**60,000+**  
SOLUTIONS, ARTICLES,  
AND DOCUMENTS  
help you find the right  
answers fast.



**10+**  
LANGUAGES  
are supported by  
Red Hat engineers and  
the Customer Portal.



**3,000**  
HOURS OF EXPERT  
TUTORIALS, TALKS,  
AND INSIGHTS  
are viewed by customers  
each year.



**60+**  
CUSTOMER PORTAL LABS  
help you configure, deploy, secure,  
and troubleshoot systems easily.

# WHERE & HOW TO START?

# Red Hat + Goodmorning Information Technology

## Discovery Session

- 고객의 주요 비즈니스 요구사항 및 목표 식별
- 비즈니스, 기술 및 운영 측면을 통한 접근
- 레드햇 전문가와 고객의 의사 결정자 및 이해 관계자 간 대화형 토론 방식으로 진행
- PoC 수행 이전 고객 환경 및 요건 검증을 위한 툴
- 고객 참여형으로 요건과 요구 사항을 사전 정의

## Design Workshop

- Discovery Session에서 도출된 개선 방안을 구체적이고 실현 가능한 Action Item을 만들어 가는 과정
- 대상 시스템 현황 분석 및 GAP분석
- 표준 아키텍처 수립 및 파일럿 업무 구현
- 중장기 플랜 (기술, 인력, 일정)
- 변화 관리 플랜 (목표, 활동, 교육 계획)

## Open Innovation Lab

- Red Hat 전문가와 함께 Agile 방법론과 DevOps 도구를 기반으로 기존 애플리케이션을 현대화 하거나 새로운 클라우드 네이티브 앱을 개발하는 몰입형 실습 포함하는 포괄적인 컨설팅
- Cultural Transformation

THANK YOU 😊