

Kubernetes, Docker, Jenkins를 활용한 CI/CD 환경 구축

삼성전자 | SQE Lab.(네트워크) | 이호광

2018. 10. 18.

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



목차

제품 특성과 현재 환경, 목표

03

DevOps

05

Immutable Infrastructure

06

Infrastructure as Code

07

Container

08

Container Orchestration

10

CI/CD

13

개선결과, Lesson Learned

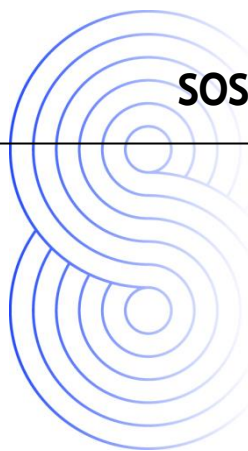
14

SOSCON 2018

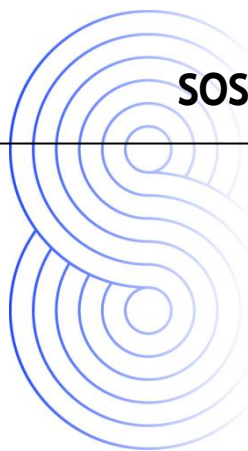
SAMSUNG OPEN SOURCE CONFERENCE 2018



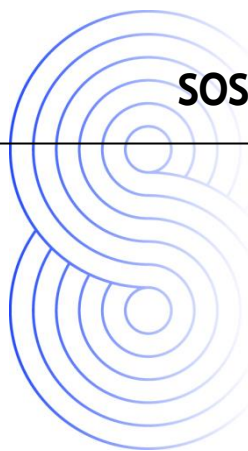
- 다양한 제품(군)과 개발/운영 중인 버전들
 - 다양한 OS와 버전
 - 다양한 toolchain
- 서버
 - 개발서버
 - 빌드서버
 - 서버관리자가 서버 세팅/관리
 - 개발자는 개발서버에서 build
- 월 평균 21,000 commits

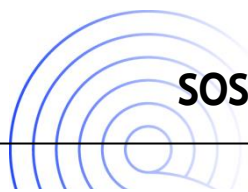


- 개발자는 개인 PC에서 개발
 - 개발서버 미제공
- 빌드서버는 Cloud Native
 - On-Premise 에서 Cloud처럼
- CI/CD Pipeline
 - Commit되면 pipeline 시작
 - Pipeline은 빌드, 시험, 배포의 일련의 workflow
 - 특정 step에서 문제 발생시 committer에게 알림
- Automation & Infrastructure as Code
 - 모든 것이 자동화
 - 누가 무엇을 바꿨는지 쉽게 알 수 있으면



- Dev(Development) + Ops (Operation)
- 개발팀과 운영팀 사이의 소통, 협업을 강조하는 S/W 개발 방법론
 - 빨리, 자주, 안정적으로 build, test, release할 수 있는 문화와 환경 구축
- 주요 Practice
 - Continuous Integration
 - 자동화된 build, 테스트 후 변경사항을 주기적으로 통합
 - Continuous Delivery
 - 코드 변경되면 자동으로 build, test, release 준비
 - Infrastructure as Code
 - Provisioning, 시스템 변경 및 구성에 대한 일관되고 반복적인 과정 자동화
 - Microservices
 - Monitoring and Logging
 - Communication and Collaboration





- 서버가 배포된 후에 수정되지 않는 인프라 패러다임
 - 변경이 필요하면 새 서버를 제공하고 기존 서버를 대체
- Snowflake Server vs Phoenix Server (2012년, Martin Fowler)
- Immutable Infrastructure의 장점
 - 일관성 및 신뢰성 향상
 - 간단하고 예측 가능한 배포 프로세스
 - 변경 가능한 인프라에서 일반적으로 발생하는 문제를 완화하거나 완전히 예방
- 효율적으로 사용하려면 다음 사항이 필요
 - 배포 자동화
 - Cloud 컴퓨팅 환경에서의 빠른 provisioning
 - 상태 저장 또는 임시 데이터 처리 솔루션

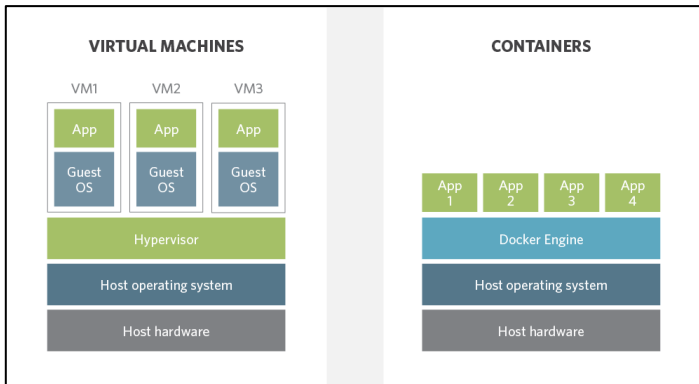


- 지금까지 수작업으로 해오던 인프라 구축이나 변경 작업을 코드로 작성해서 자동화하는 것
- 인프라에 "유연성", 즉 반복성과 확장성을 부여
- 장점 : 빠른 속도, 비용 절감, 위험 감소
- Tools
 - Ansible
 - Chef
 - Puppet
 - Terraform
 - AWS CloudFormation

```
1 FROM alpine:3.8
2
3 RUN apk add --no-cache \
4     ca-certificates
5
6 # set up nsswitch.conf for Go's "netgo" implementation (which Docker explicitly uses)
7 # - https://github.com/docker/docker-ce/blob/v17.09.0-ce/components/engine/hack/make.sh#L149
8 # - https://github.com/golang/go/blob/go1.9.1/src/net/conf.go#L194-L275
9 # - docker run --rm debian:stretch grep '^hosts:' /etc/nsswitch.conf
10 RUN [ ! -e /etc/nsswitch.conf ] && echo 'hosts: files dns' > /etc/nsswitch.conf
11
12 ENV DOCKER_CHANNEL stable
13 ENV DOCKER_VERSION 18.06.1-ce
14 # TODO ENV DOCKER_SHA256
15 # https://github.com/docker/docker-ce/blob/5b073ee2cf564edee5adca05ee574142f7627bb/components/package
16 # (no SHA file artifacts on download.docker.com yet as of 2017-06-07 though)
17
18 RUN set -ex; \
19 # why we use "curl" instead of "wget":
20 # + wget -O docker.tgz https://download.docker.com/linux/static/stable/x86_64/docker-17.03.1-ce.tgz
21 # Connecting to download.docker.com (54.230.87.253:443)
22 # wget: error getting response: Connection reset by peer
23     apk add --no-cache --virtual .fetch-deps \
24         curl \
25         tar \
26     ; \
```

<Docker in Docker!> (출처 : <https://github.com/docker-library/docker>)

- Container 란?
 - VM과 유사하나, Host OS의 Kernel을 공유하면서 애플리케이션과 라이브러리만 가상화하는 방법

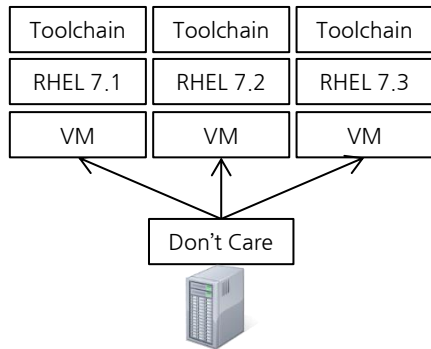


(출처 : techtarget.com)

- Container 기술의 장점
 - 가볍다 : VM을 사용할 때보다 최대 4~10배나 더 많은 서버 인스턴스 구동 가능
 - 빠르다 : 컨테이너가 구동되면 애플리케이션이 바로 시작됨

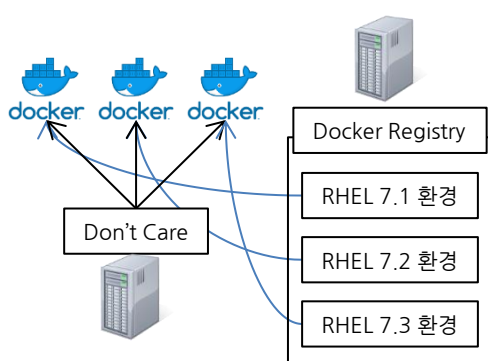
- Container 기술 중 현재 가장 인기 있는 오픈소스 소프트웨어
- 표준 컨테이너 포맷 지위 획득 (2015년 OCI(Open Container Initiative)에 소스 기증)
- 이미지 생성과 배포/관리에 특화되어, Cloud와 같은 유연성을 제공
- 비저장성 - 영속성(Persistency) 없음 (필요 시 별도 volume 사용)

As-Is

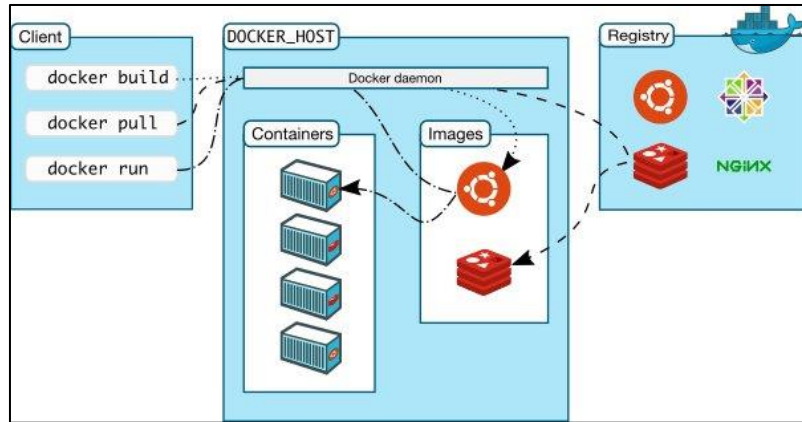


- VM 개수만큼 각종 OS, Toolchain 설치
- 환경이 변경되면, VM 개수만큼 작업

To-Be

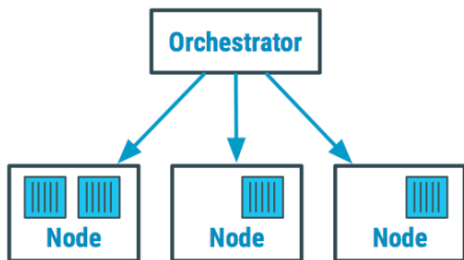


- 이미지 관리 (환경당 1회)
- 환경이 변경되면 이미지 업데이트

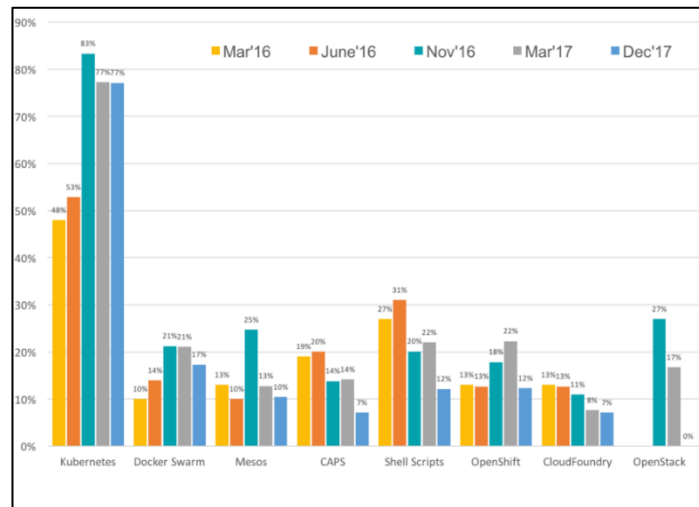


(출처 : docker.com)

- 수많은 호스트에서 다양한 container를 세팅하고 적용/관리해야 하는 번거로움을 해결
- 컨테이너의 배포, 관리, 확장, 네트워킹 및 가용성을 자동화하는 도구
- 기능
 - 호스트 Provisioning
 - 컨테이너 세트의 인스턴스화
 - 실패한 컨테이너 일정 재조정
 - 합의된 인터페이스를 통해 컨테이너 연결
 - 클러스터 외부에 서비스 노출
 - 컨테이너 추가 또는 제거로 클러스터 확장 또는 축소



(출처 : docker.com)

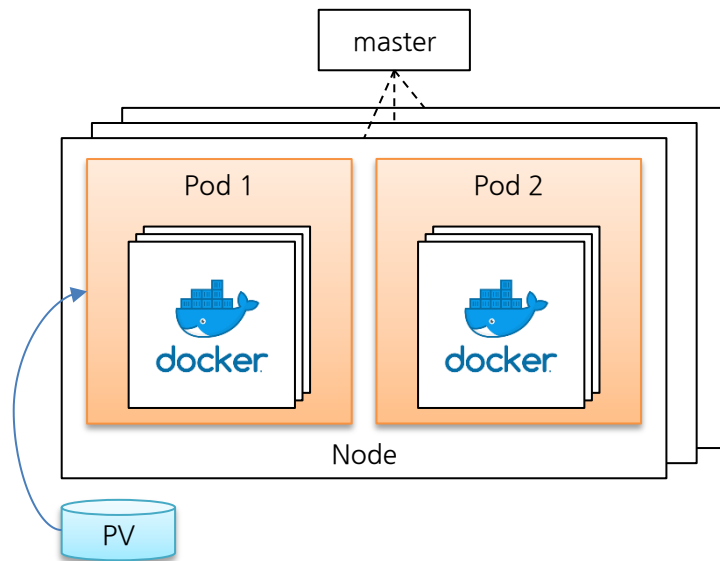


<The Container Orchestration Landscape is Changing> (출처: cncf.io)

- Container Orchestration 도구 중 가장 인기 있는 오픈소스 소프트웨어
- Google이 자사 데이터센터 운영에 활용했던 컨테이너 기술을 기반으로 개발해 오픈소스로 공개
- CNCF(Cloud Native Computing Foundation)에 2016년 첫번째 프로젝트로 기부



(출처 : cloudplatform.googleblog.com)



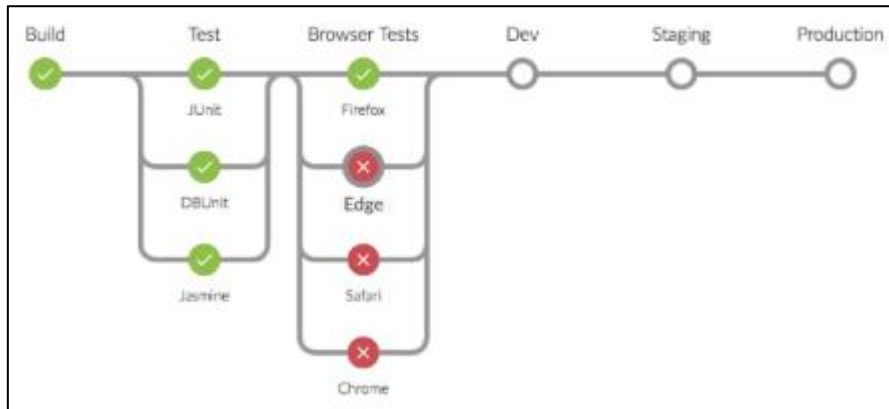
- Harbor  HARBOR™
 - VMware사에서 만든 오픈소스 Docker registry
 - 현재 CNCF sandbox project
 - Helm 
 - The package manager for Kubernetes
 - chart 활용하여 복잡한 Kubernetes application을 쉽게 정의, 설치, 업그레이드할 수 있음
 - chart를 생성, 버전 관리, 공유, 배포하기 쉬움
 - 현재 CNCF Incubating project
- 

- Fail Fast
- CI (Continuous Integration) (1997년, Kent Beck)
 - change가 commit되면 즉시 build & test
 - 가능한 빠른 오류 발견과 수정을 가능케 한다
 - 빨리 고품질의 code를 만들 수 있다
- (2010년, Jez Humble)
 - CD (Continuous Delivery or Continuous Deployment)
 - CI를 더 오른쪽으로(test, deploy) 확장하여 release-ready 상태 보장
 - 빠르고 반복적이고 믿을 수 있는 방법으로 change가 pre-production 또는 production 환경, 결국 고객에게 까지 전달될 수 있게 한다
 - Continuous Deployment는 Facebook, Netflix와 같은 B2C 제품에 적합





- Cloudbees사에서 만든 오픈소스 CI/CD Tool
- 다양한 Plugin 제공
- Pipeline as Code

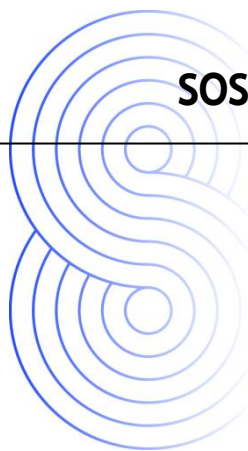


〈Blue Ocean에서의 Pipeline〉 (출처 : Jenkins.io)

```
pipeline {
  agent any
  stages {
    stage('Build') {
      steps {
        //
      }
    }
    stage('Test') {
      steps {
        //
      }
    }
    stage('Deploy') {
      steps {
        //
      }
    }
  }
}
```

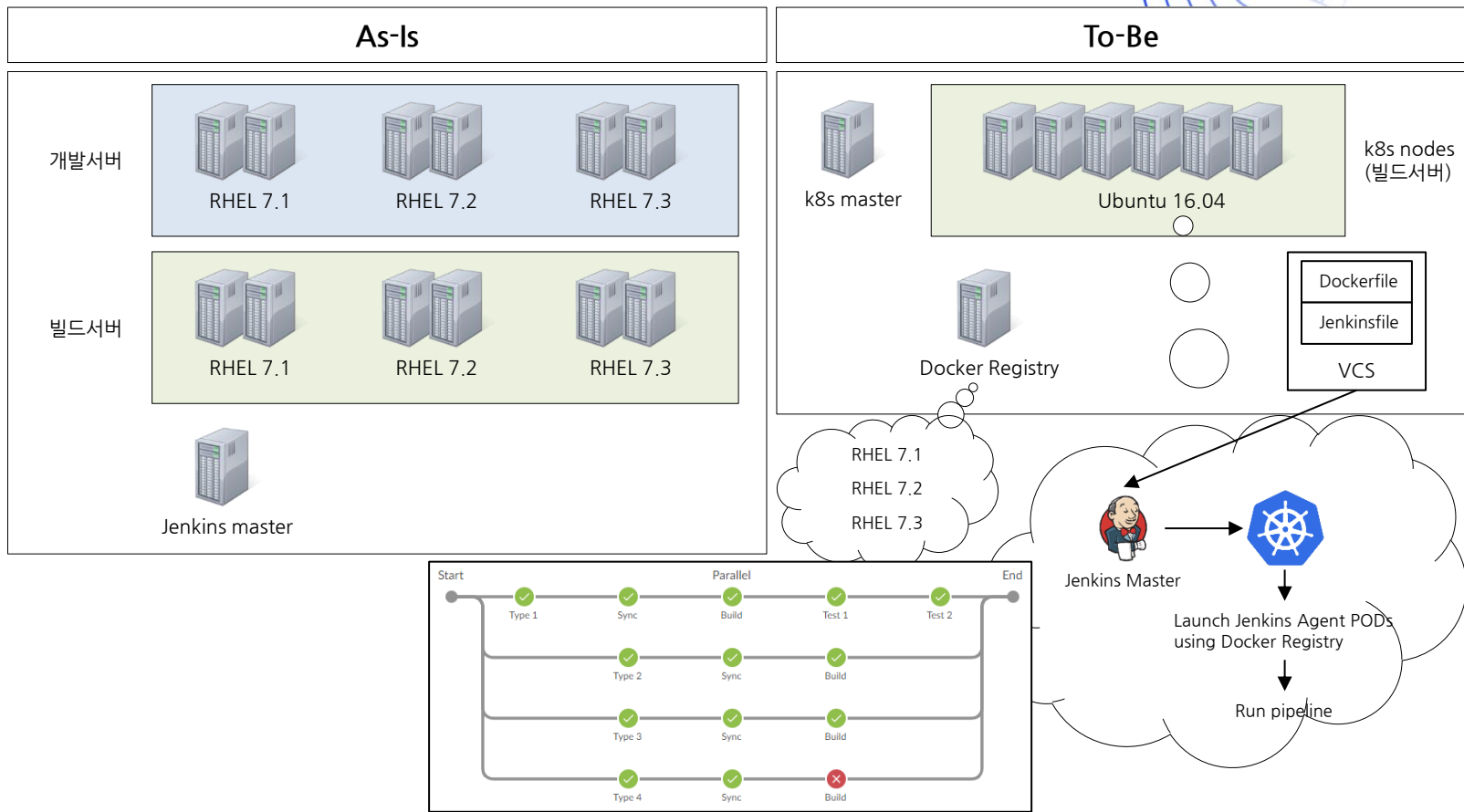
〈Pipeline as Code : Jenkinsfile〉 (출처 : Jenkins.io)

- 개발환경 Container화
 - 개발환경을 Dockerfile로 작성하고 형상관리
 - Harbor로 Docker Registry 구축하고 이미지 등록
 - 개발서버 대신 개인PC에서 개발
- Kubernetes 활용한 Cloud Native 환경 구축
 - kubeadm 활용한 Kubernetes Cluster 구축
 - Ingress, Persistent volume, Helm 설치/활용
 - Jenkins 설치
- Jenkins 운영
 - Blue Ocean, Kubernetes, Pipeline 등의 Plugin 활용
 - MultiBranch Pipeline Job 생성
 - 제품/branch별 Jenkinsfile 작성하고 형상관리



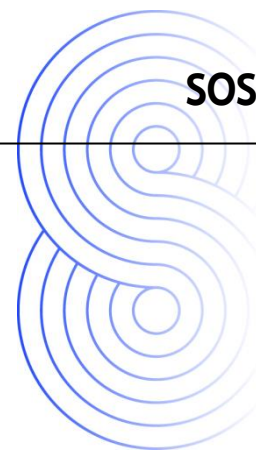
개선 결과

SOSCON 2018



- Infra 기술들이 Cloud를 향하고 있다 (또는 Cloud의 대중화)
 - Hybrid (Cloud + On-premise) 환경으로 전환 예상
- DevOps Engineer ?
- Docker (+ Harbor)
 - Dockerfile 작성/관리 중요
- Kubernetes (+ Helm)
 - 난이도 상 (http(s) proxy 문제 등)
 - 아직 완성되지 않았지만 (beta 버전) 강력하다
- Jenkins
 - 설정 자동화 필요 (Configuration as Code Plugin 활용)





THANK YOU

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

