

2022.11.17(Thu)

# Cloud Native와 DevOps

우수연 / IBM

# SPEAKER

## Suyeon Woo (우수연)

---

- Technology Engineer @IBM
- AWS Community Builder #containers
- AUSG Co-Organizer  
(AWSKRUG University Student Group)
- GDG Cloud Korea Organizer  
(Google Developer Group)

#Cloud #Data #Web3.0



# AGENDA

Cloud Native  
DevOps (feat. SRE)

- ✓ 개발 프로젝트를 하나 이상 완성해본 적이 있다.
- ✓ Cloud에 대해 설명할 수 있다.
- ✓ Cloud Native에 대해 들어본 적 있다.
- ✓ DevOps 엔지니어를 고민중이다.

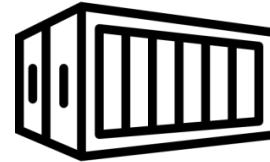
# Paradigm



Physical Machine  
(Colocation)



Virtual Machine



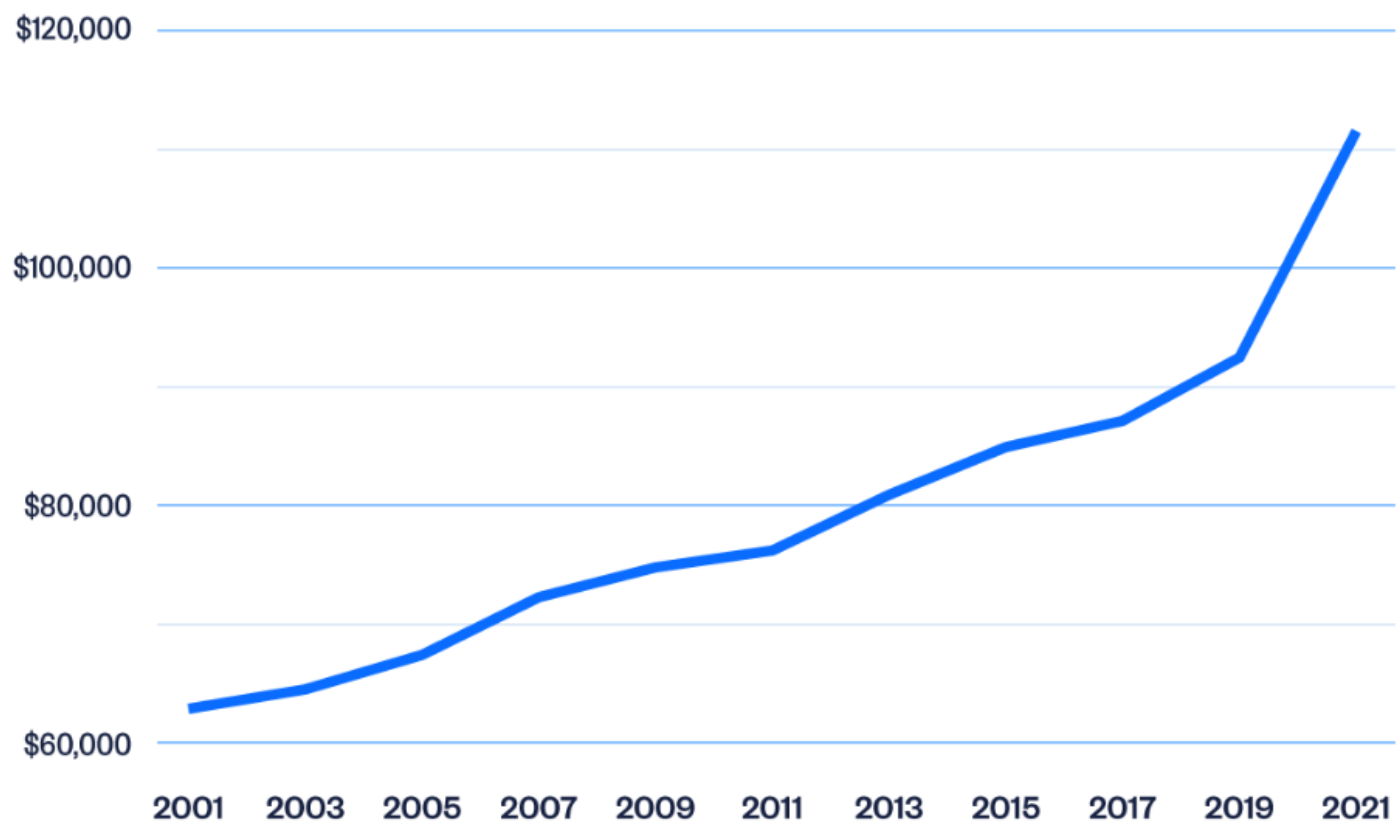
Containerization



Serverless

Abstraction / Managed

Focus on the Business



오늘날 개발자가 각광받는 이유??

# Use of Cloud-Native Technologies Will Be Pervasive, not Just Popular

By 2025, Gartner estimates that over 95% of new digital workloads will be deployed on cloud-native platforms, up from 30% in 2021.

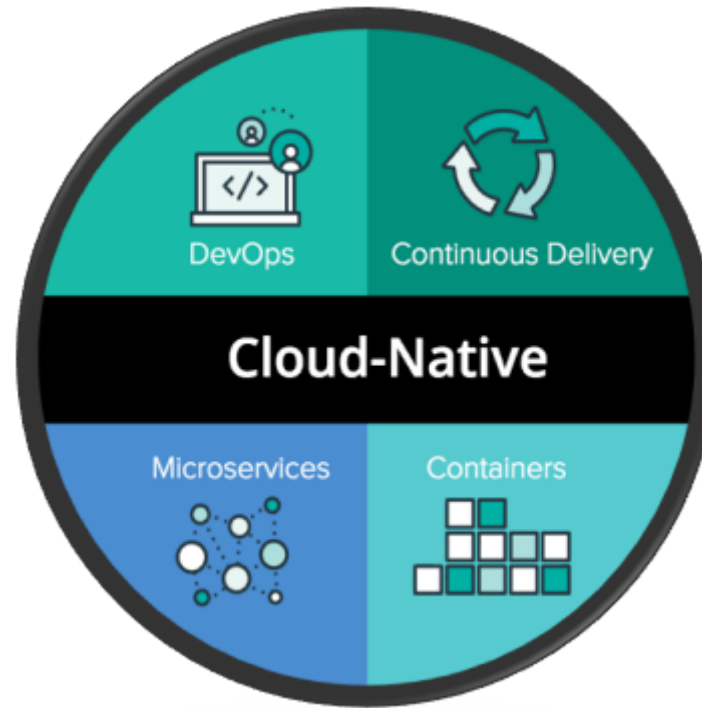
# Low-Code and No-Code Technologies Use Will Nearly Triple by 2025

By 2025, 70% of new applications developed by organizations will use low-code or no-code technologies, up from less than 25% in 2020.

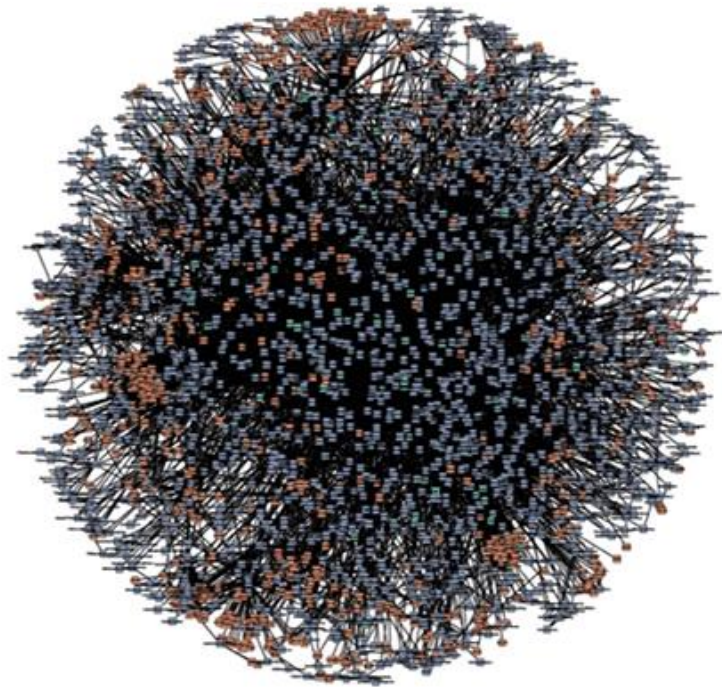


**Cloud Native**

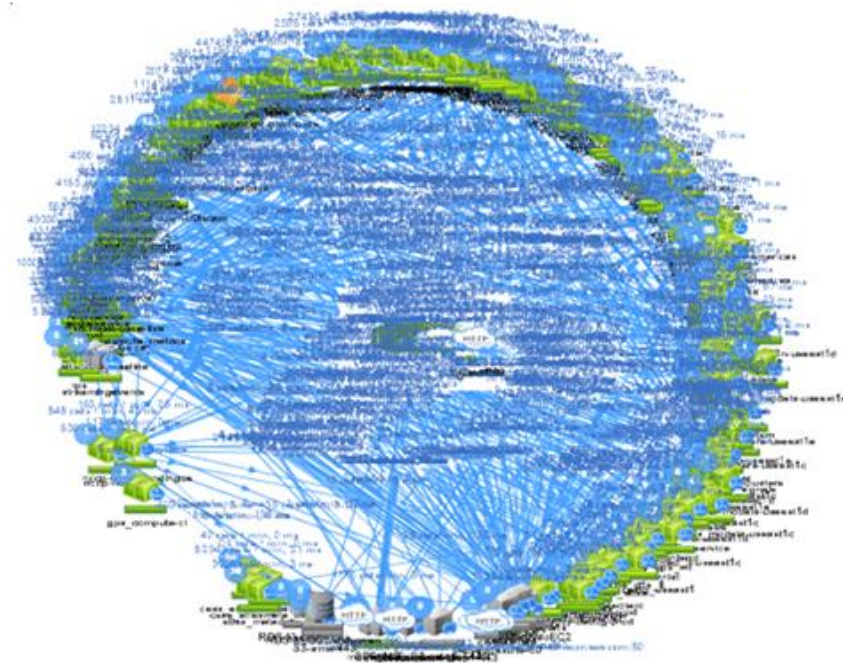
# Cloud Native 주요 요소



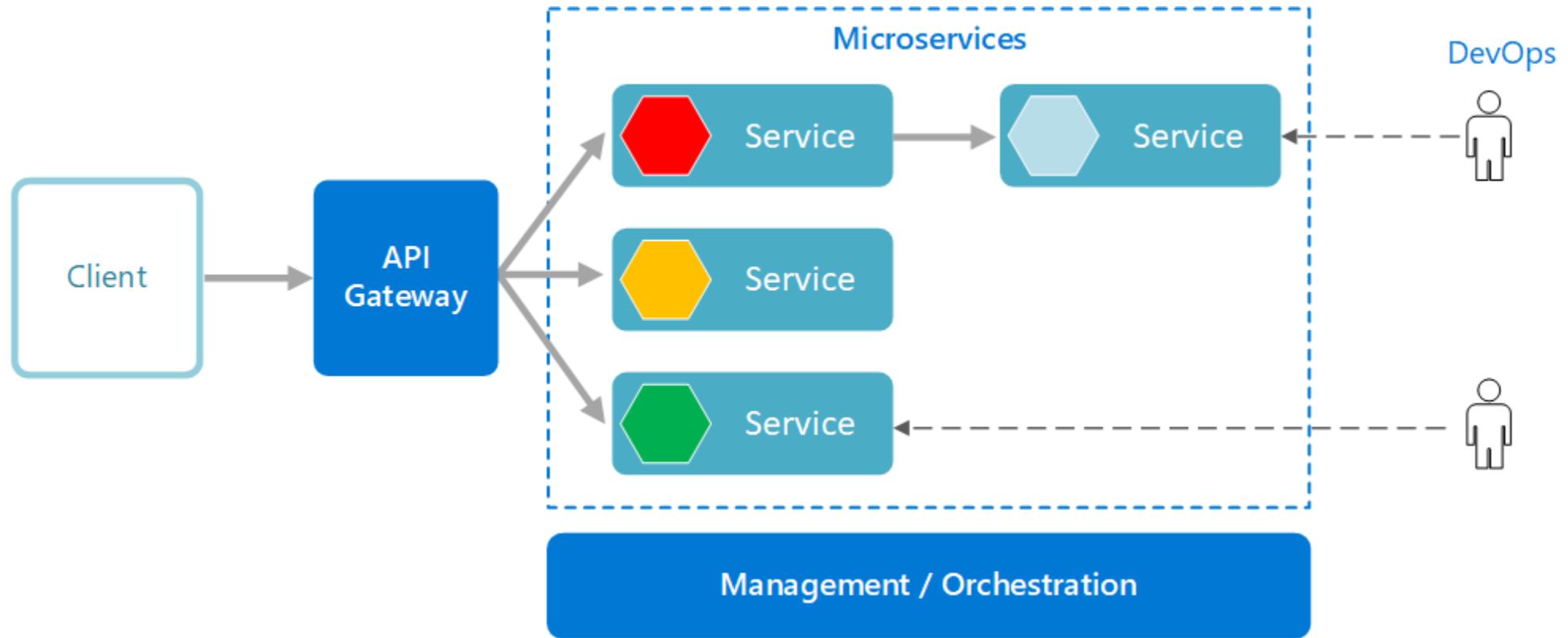
# 1. Microservice



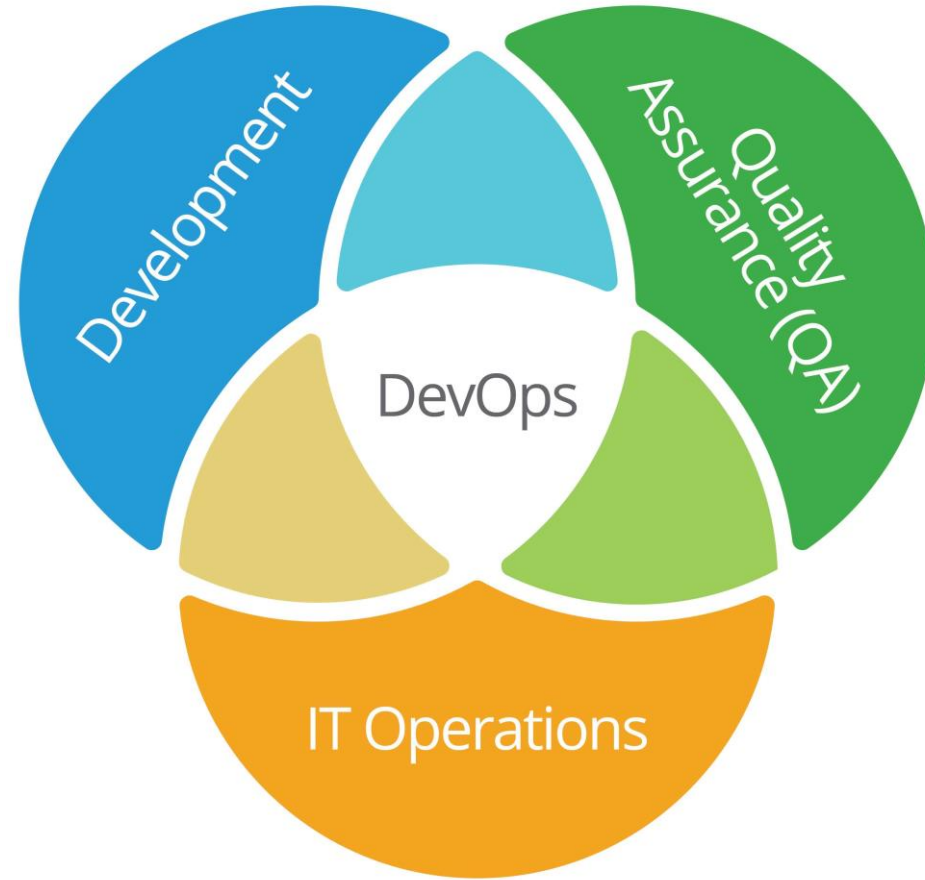
amazon.com®



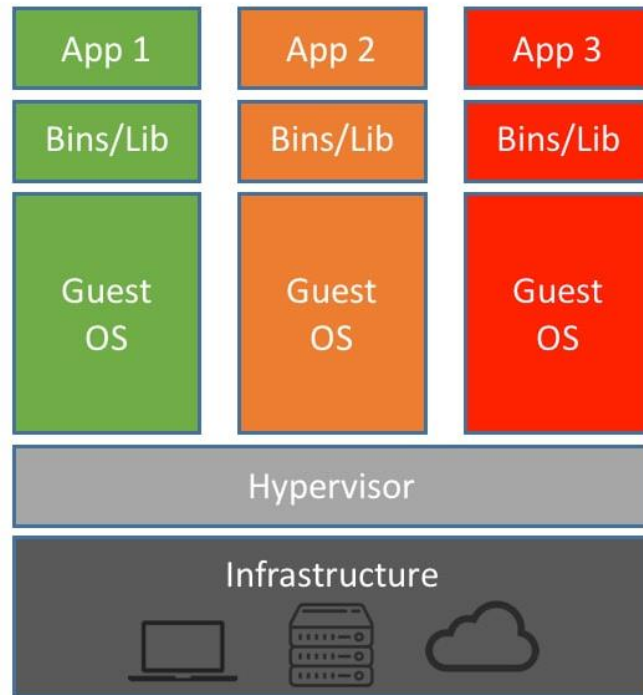
# 1. Microservice



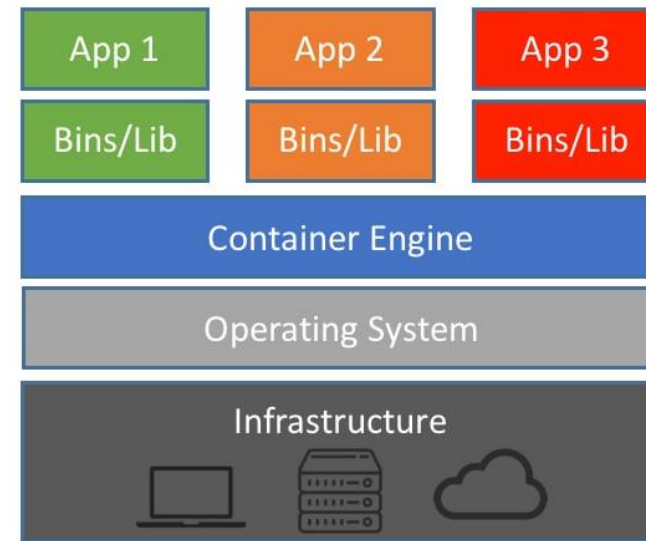
## 2. DevOps



# 3. Container

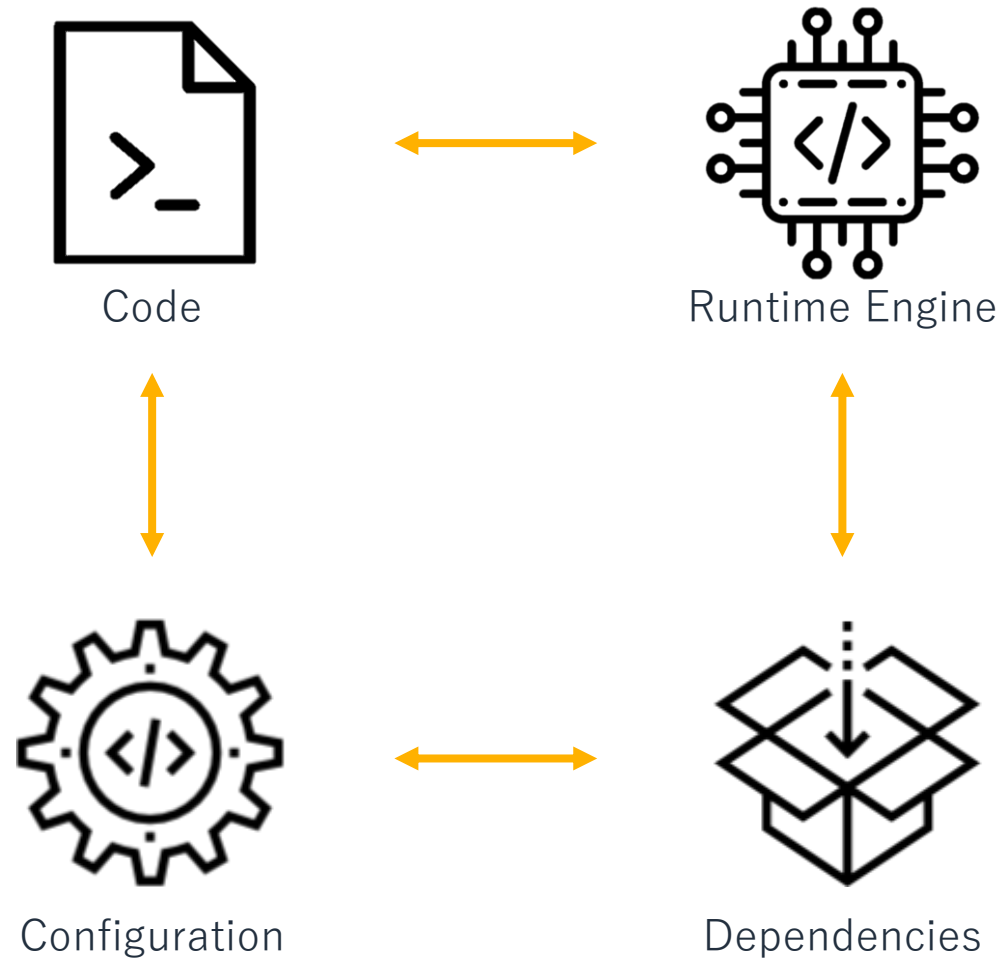


Virtual Machine

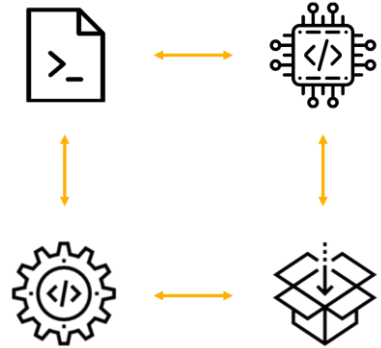


Container

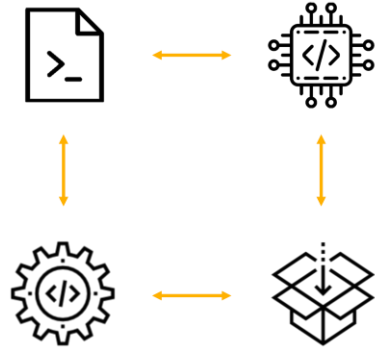
# 3. Container



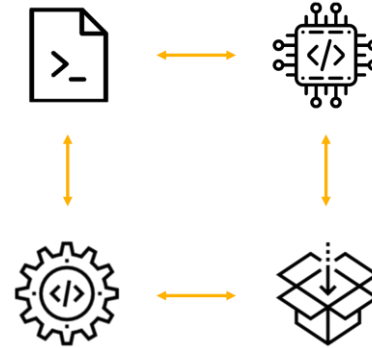
# 3. Container



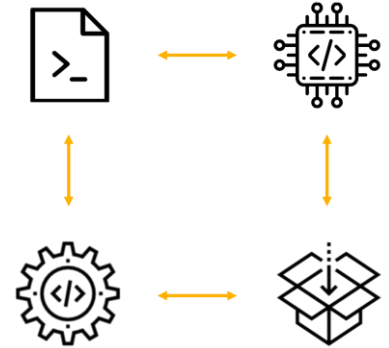
Local PC



DEV



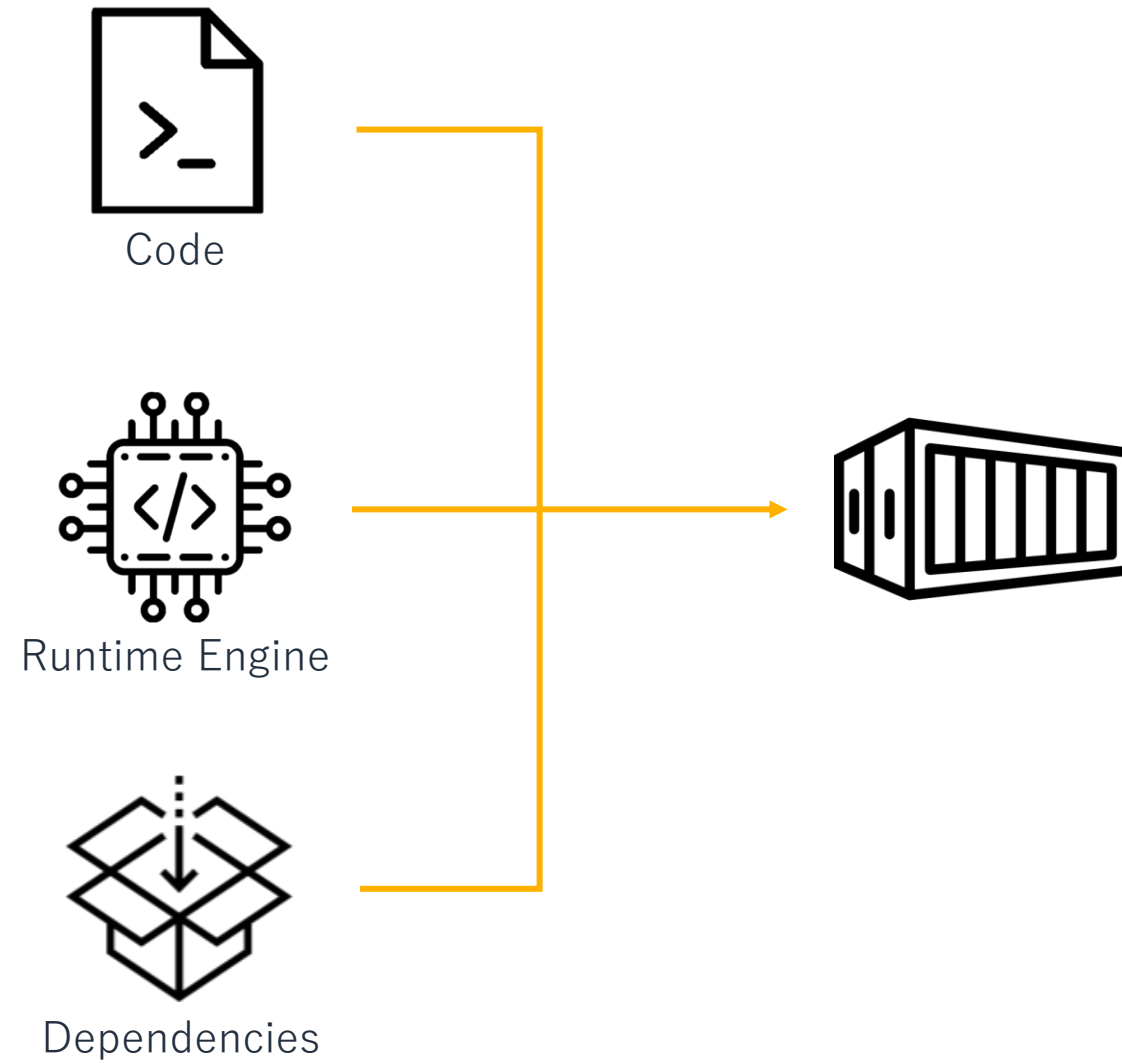
STAG



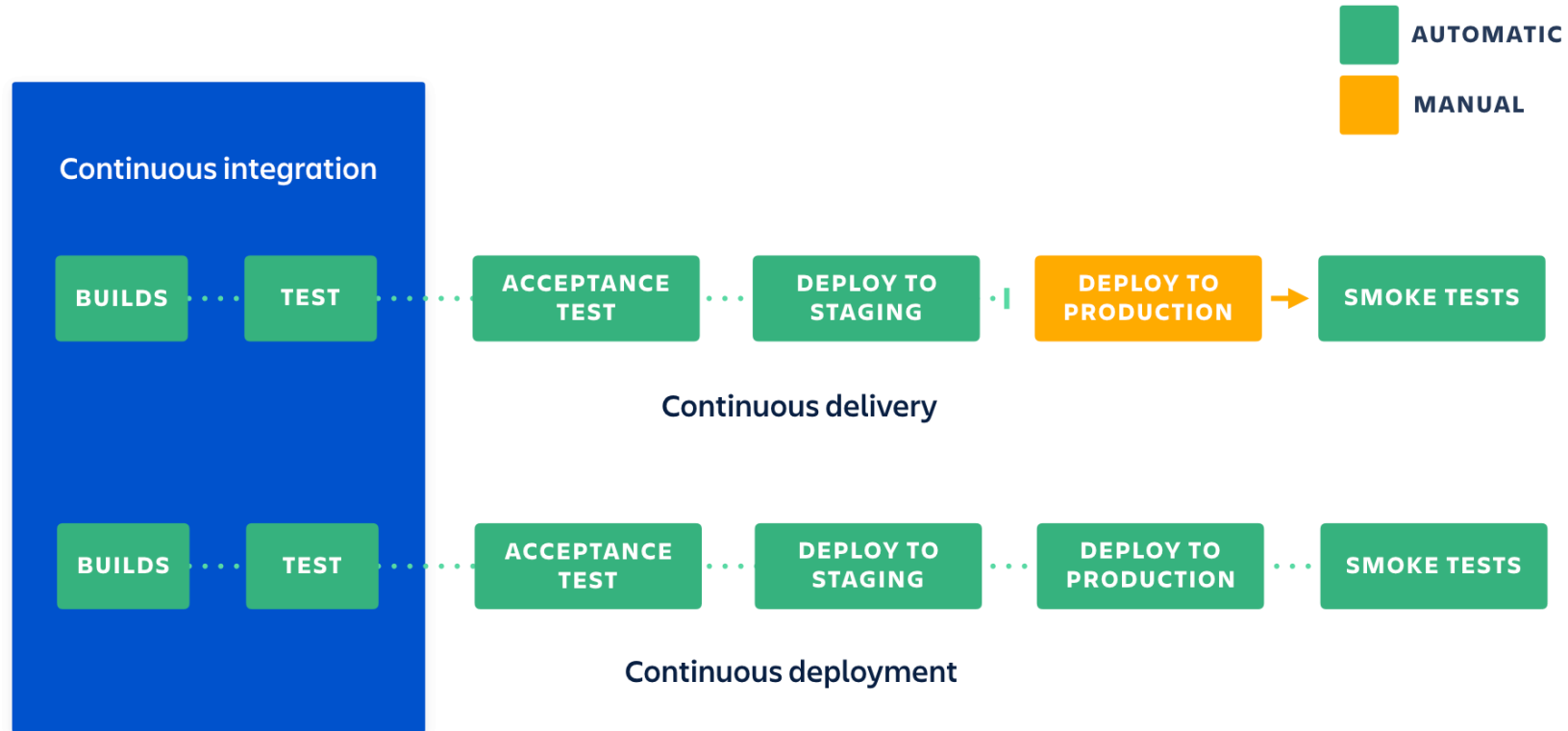
PROD













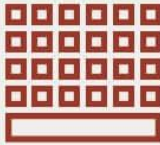

# 3. Container



# 4. Continuous Delivery



# Cloud Native Architecture

	Development Process	Application Architecture	Deployment & Packaging	Application Infrastructure
~ 1980 ~ 1990	Waterfall 	Monolithic 	Physical Server 	Datacenter 
~ 2000	Agile 	N-Tier 	Virtual Servers 	Hosted 
~ 2010	DevOps 	Microservices 	Containers 	Cloud 

# WHY? Cloud Native??



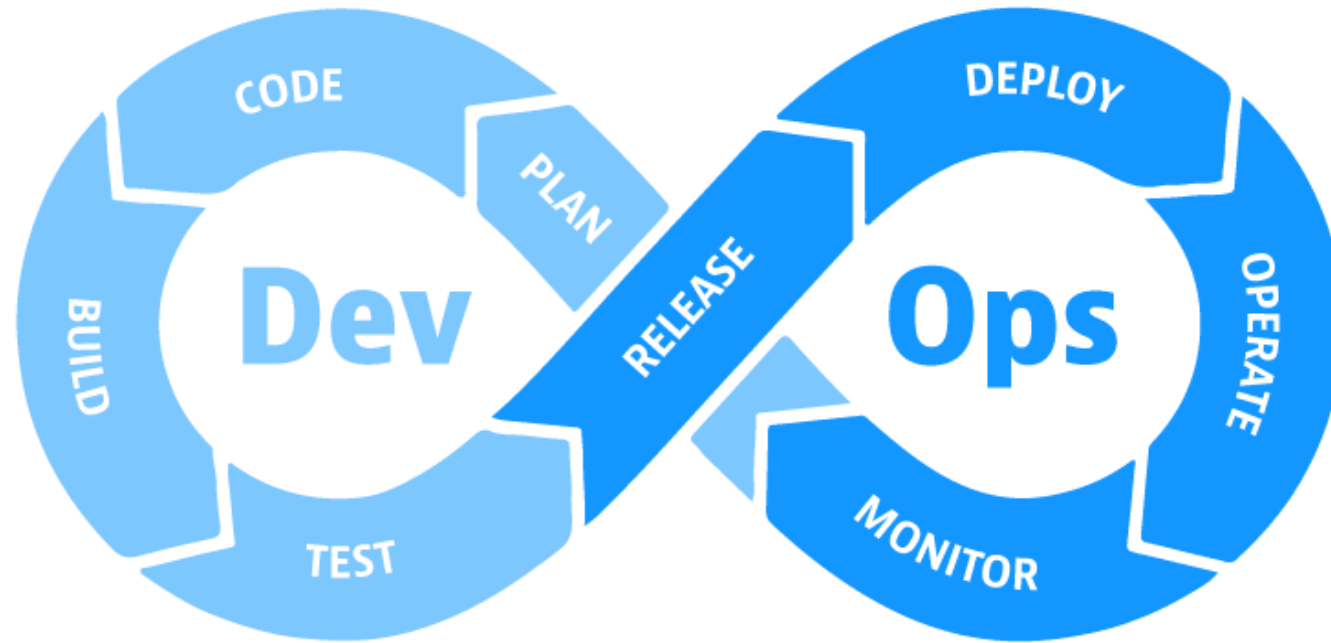
# **DevOps (feat. SRE)**

# DevOps

**Dev**elopment + **Op**erations

# Development + Operations

➤ Development : 개발팀, Operations : 운영팀??



# 개발팀 vs 운영팀

## ▪ 개발팀

- 사용자의 요구사항 반영
- 최소 기능 단위 빠른 배포
- 비즈니스의 성장
- 개발 생산성 향상
- 최신 기술 사용

## ▪ 운영팀

- 운영에 대한 책임
- 보안 리스크 최소화
- 안정성



# 그래서 DevOps가 먼데?

- DevOps는 방법론이 아니다.
- DevOps는 소프트웨어가 아니다.
- DevOps는 팀 혹은 직무가 아니다.

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams.

- Atlassian -

DevOps는 소프트웨어 개발 팀과 IT 팀 간의 프로세스를 자동화하고 통합하는 일련의 관행, 도구 및 문화적 철학이다.

**DevOps is the union of people, process, and products  
to enable the continuous delivery of value to our end-users.**

**- Microsoft -**

DevOps는 최종 사용자에게 지속적인 가치 제공을 가능하게 하는 사람, 프로세스 및 제품의 결합이다.

**DevOps**는 문화, 철학 또는 사고방식일 뿐  
구체적인 방법을 제시하지 않음

**그럼 어떻게??**

**Site Reliability Engineering**

“ class **SRE** implements **DevOps** ”

**Site Reliability Engineering**



google.com



시스템 신뢰성(안정성)



소프트웨어 엔지니어링

INFCON 2022

배우고 나누고 성장하세요.



AWS로 알아보는 서비스  
아키텍처의 변화와 진화

LGU+ 송주영

Inflearn

## DevOps 업무



서비스 개발과 배포를 신속하고 정확하게 만드는 것



서비스와 팀의 문제를 빠르고 정확하게 감지하는 것



장애를 최대한 빠른시간안에 해결하는 것



유연하고 신뢰할만한 아키텍처를 만드는 것



데이터에 기반한 원활한 의사소통이 가능하게 하는 것

# 구체적인 가이드라인 : SRE 엔지니어가 하는 일



## Metrics & Monitoring

- SLOs
- Dashboards
- Analytics



## Capacity Planning

- Forecasting
- Demand-driven
- Performance



## Change Management

- Release process
- Consulting design
- Automation



## Emergency Response

- Oncall
- Analysis
- Postmortems



## Culture

- Toil management
- Engineering alignment
- Blamelessness



# 구체적인 가이드라인 : SRE 엔지니어가 하는 일

## 1. Metrics & Monitoring

- 모니터링 지표 정의
- 메트릭 시각화
- 지표 분석 후 인사이트 도출

→ 운영 안정성

# 구체적인 가이드라인 : SRE 엔지니어가 하는 일

## 2. Capacity Planning

- 시스템 자원 확보
- 소프트웨어 최적화

→ 소프트웨어 안정성

# 구체적인 가이드라인 : SRE 엔지니어가 하는 일

## 3. Change Management

- 소프트웨어 배포
- 인프라 변경 / 시스템 설정 변경
- 휴먼 에러 최소화

→ 프로세스 안정성

# 구체적인 가이드라인 : SRE 엔지니어가 하는 일

## 4. Emergency Response

- 장애 처리
- 신속한 복구

→ 시스템 안정성

# 구체적인 가이드라인 : SRE 엔지니어가 하는 일

## 5. Culture

- DevOps 문화를 만들고 지켜 나가기
- 문화를 만들기 위한 방법
  - a. 데이터에 기반한 합리적인 의사결정
  - b. 장애에 대한 비난 X, 분석하고 자동화 O
  - c. 책임 나눠 가지기

## DevOps 문화가 잘 정착된 팀은?

1. 더 자주 배포
2. 커밋 ~ 배포까지의 Leadtime 단축
3. 실패율 감소
4. 사고로부터 신속한 복구

**Q & A**

# Thank you!

 dntndus9611@gmail.com

 suyeon-woo

 sye0n\_w