

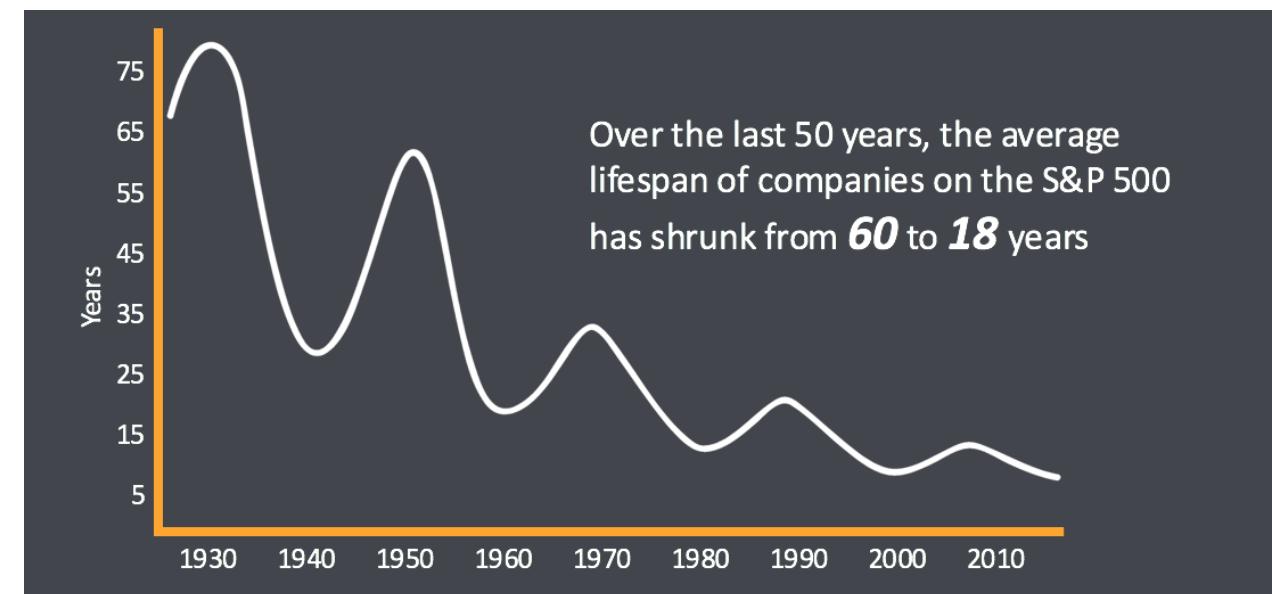
Cloud Native & DevOps 개요

Mee-Nam Lee
Oracle Middleware



So Why Cloud Native? Business and Technology Drivers

- Speed – deliver ahead of biz needs
- Agility – do more with less
- Cost – optimize resources & systems
- Continuously innovate or perish
 - Uber / Cabs ; Netflix / Video rentals
- Continuous user experience
 - Not one & done, evolving



Continuous innovation and deployment of fresh software is one of the few measures that will keep a company from becoming the next to drop off the list.

What Do Today's Winners Have in Common?



Speed of Business Innovation, Enabled by Software

혁신과 차별화를 위한 소프트웨어는 “개발”된다

75%

By 2020, 75 percent of application purchases supporting digital business will be "build," not "buy."

Forecast Analysis: Enterprise Application Software, Worldwide, 2Q15 Update.

Gartner®

Buzzwords You May Hear

Key terms and methods around Cloud Native

Microservices

DevOps

APIs/API-First/Mobile-first

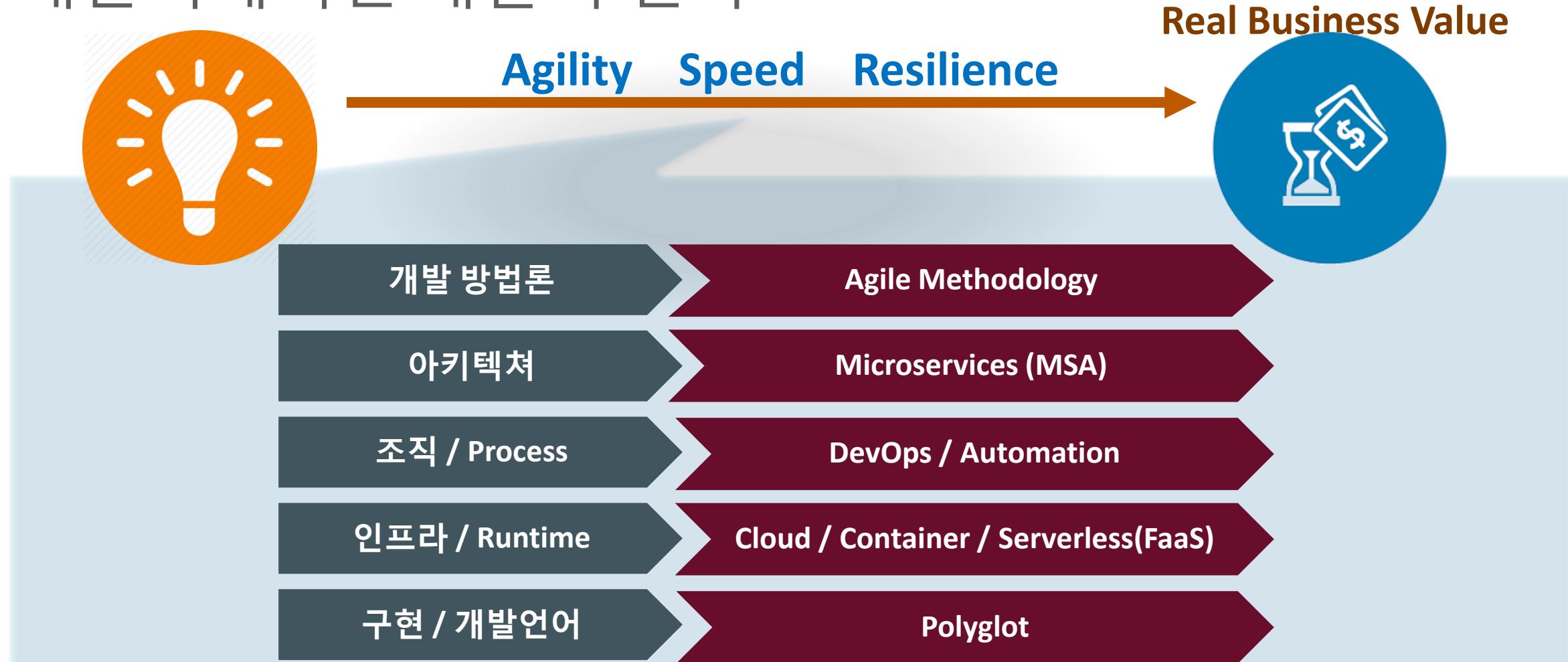
Containers/Docker

Polyglot



12-factor

애플리케이션 개발의 변화



간단한 Cloud Native 예시

모든 것이 서비스로서 제공되고, 커스터마이징된 전체를 만들기 위해 조합됨

The screenshot shows the Alibaba.com website with a product listing for a quartz watch. The product is described as "NEW! JAPAN MOVT QUARTZ WATCH STAINLESS STEEL BACK, WATCHES FOR MEN". Key details include:

- FOB Price: US \$15 - 20 / Piece | Get Latest Price
- Min.Order Quantity: 500 Piece/Pieces
- Supply Ability: 3000 Piece/Pieces per Day
- Port: SHENZHEN
- Payment Terms: L/C,T/T,Western Union,MoneyGram

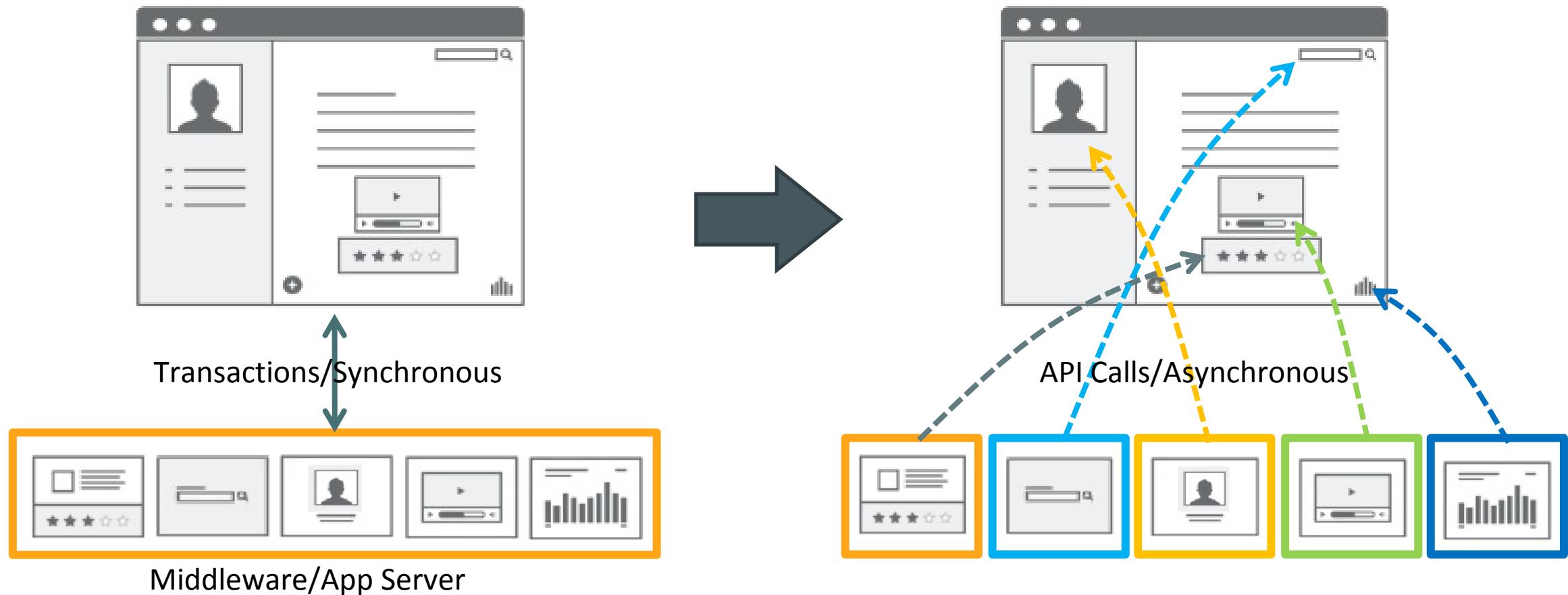
Call-to-action buttons include "Contact Supplier", "Start Order", and "Request Sample". The page also highlights "Trade Assurance" and provides payment protection guarantees.

On the right, a hand holds a smartphone displaying the same product listing on the Alibaba mobile app, showing the responsive design of the platform.

Header navigation includes: Sourcing Solutions, Services & Membership, Help & Community, Selection, AliSource, Wholesaler, Categories, Products, What are you looking for..., Search, Sign In, Join Free, My Alibaba, Order Protection, Multi-Language Sites, and Home > Products > Timepieces, Jewelry, Eyewear > Watches > Wristwatches (171846).

Bottom footer includes: ORACLE, See larger image, ZOOM, http://www.ydwatch.com, Trade Assurance, Place order online and pay to the designated bank account to get full protection., 100% Payment protection, 100% On-time shipment, 100% Product quality protection, Add to My Favorites, and Share icon.

Microservices



Reference: <http://www.slideshare.net/RichardHarvey7/micro-services-and-containers>

Microservices란?

독립적으로 배포되는 최소화된 기능의 서비스지만 광범위한 사용사례를 위해 서로 상호작용하는 서비스

Monolithic Applications

- 단일, Monolithic App
- 전체 App을 함께 배포해야 함
- 전체 App이 단일 DB 사용
- 기술 계층 중심으로 구성
- 각각의 런타임 인스턴스에 상태를 보관
- 전체 App이 하나의 기술 스택을 채택
- SOAP 사용



Microservices

- 다수의, 더 작아진 최소 기능의 Microservices
- 각 Microservice를 독립적으로 배포
- 각 Microservice는 각자의 데이터 저장소를 가짐
- 비즈니스 기능 중심으로 구성
- 상태는 외부에 보관
- 각 Microservice가 각자의 기술을 선택
- HTTP를 통한 REST, Messaging, 또는 Binary

You're Doing Microservices If You...

Can build a
microservice
independently

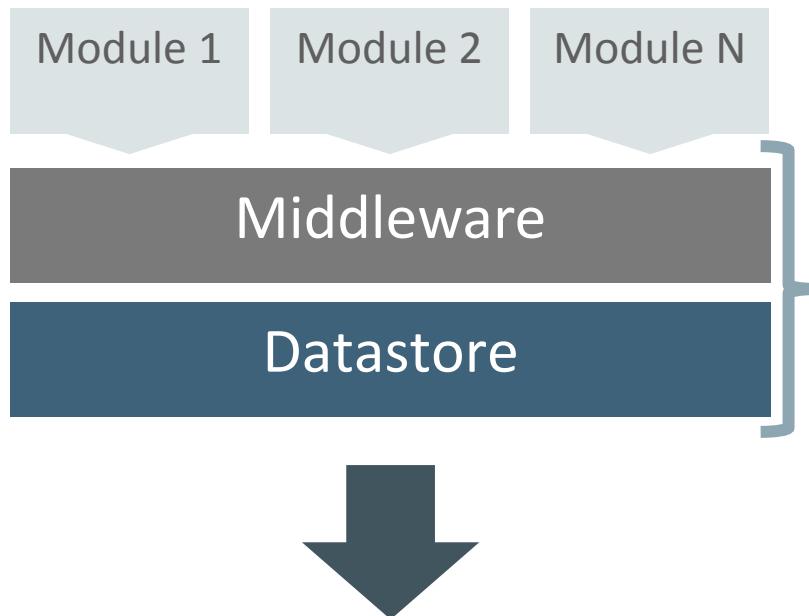
Can release each
microservice
independently

Don't share a
datasource across
microservices

“Micro” in Microservices != Runtime Weight

애플리케이션을 많은 작은 조각으로 나누어 마켓에 필요한 기능을 신속하게 획득

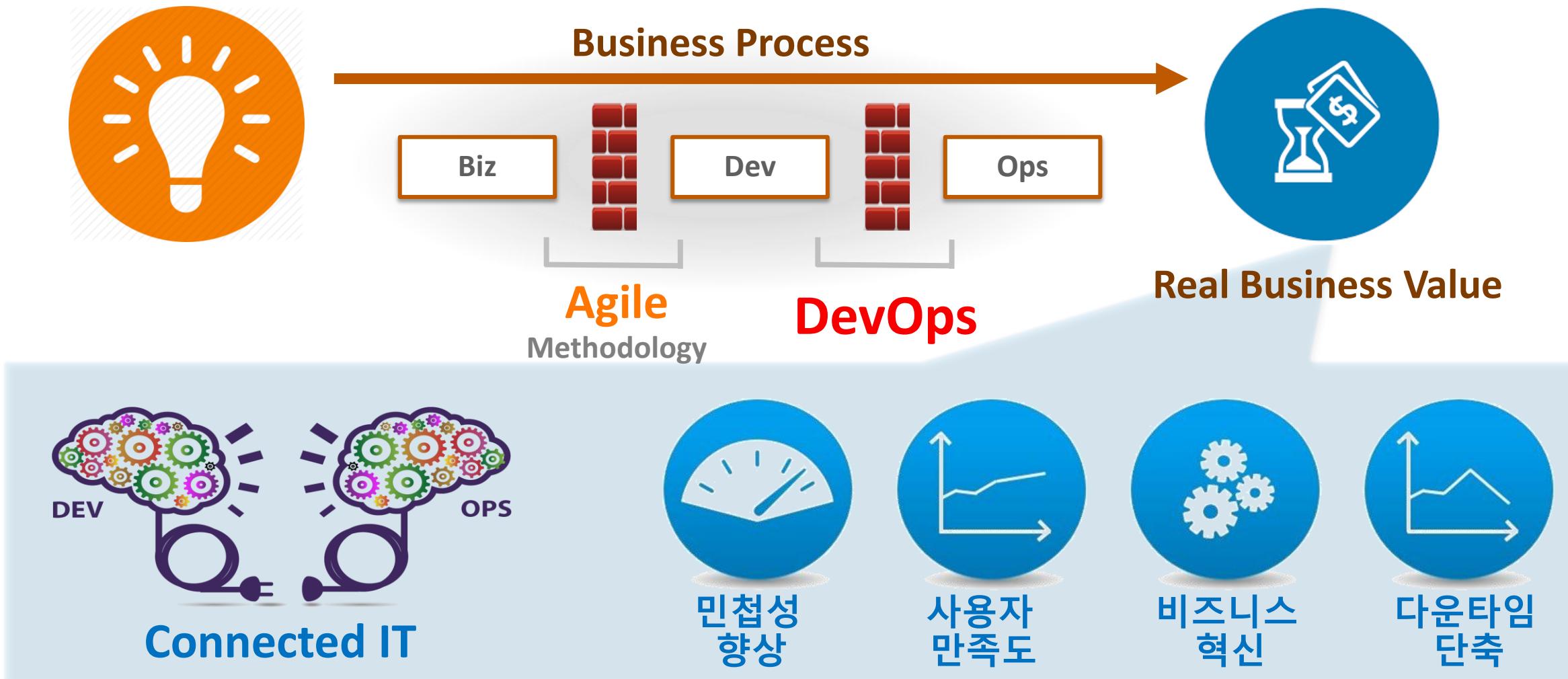
Monoliths



Microservices



왜 DevOps 인가?



익숙한가요?

DevOps는 이런 문제를 해결합니다



내 시스템은
문제 없어.
애플리케이션
이 문제야!

내 코드는 문제
없어. 시스템
이슈야!

Dev and Ops Constantly Argue

“Code is written...it’s your problem now”



What are the Core DevOps Principles?

Cultural movement enabled by technology

Dev

Ops

Paid to add new features

Paid to keep system stable, fast and available

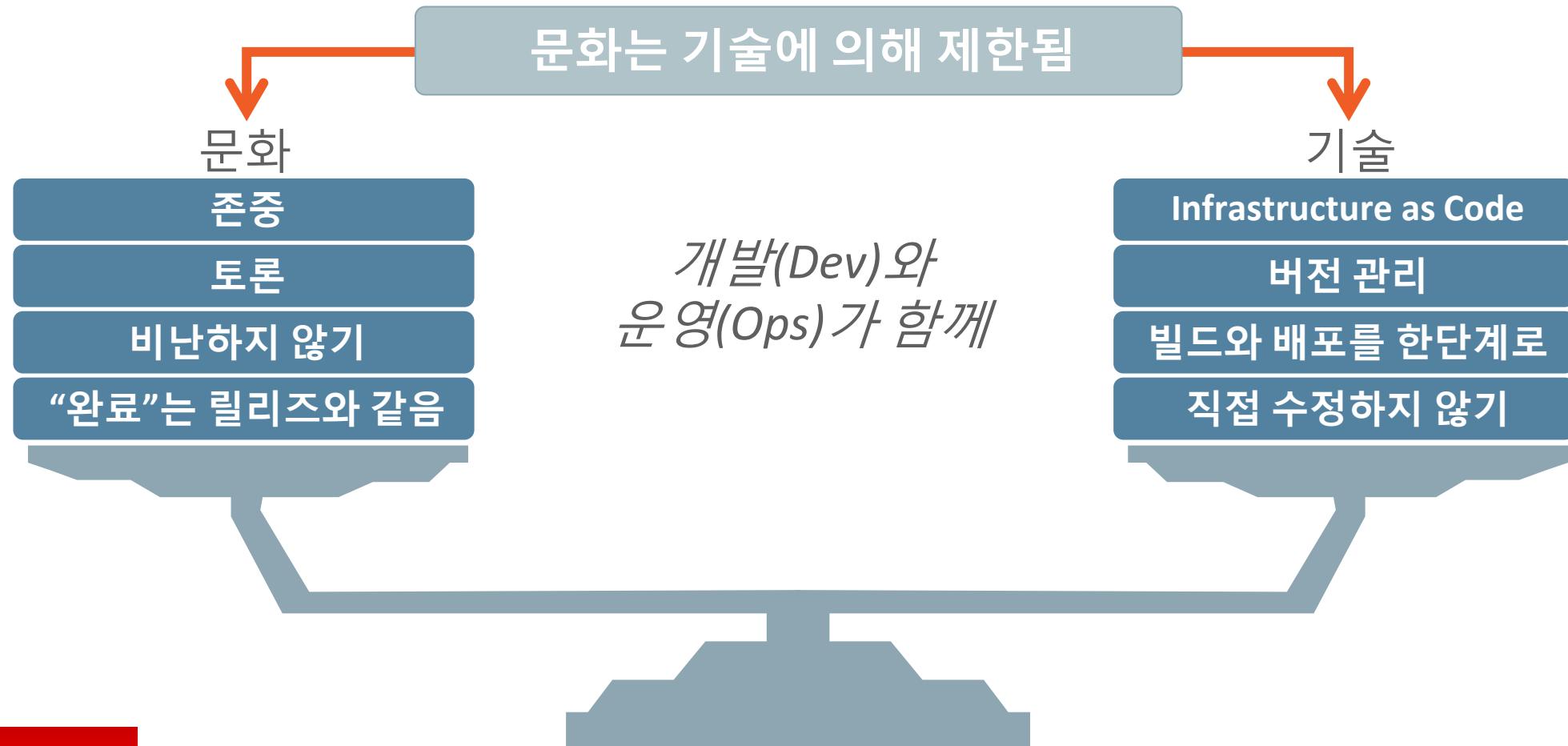
New goal:

Add new features
FAST and keep the system stable, fast and available

DevOps

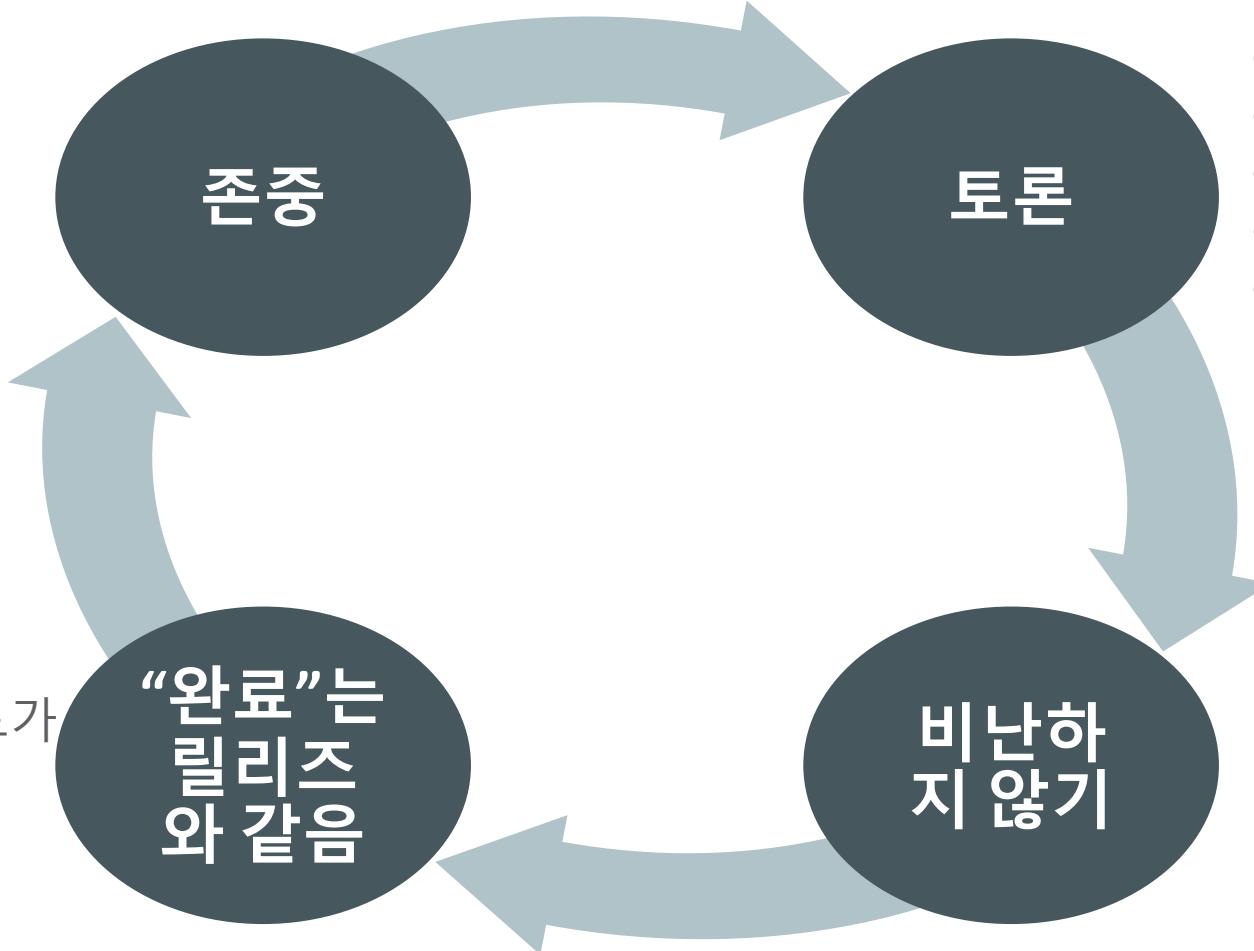
DevOps

문화와 기술의 균형



DevOps 원칙 #1: 문화

- Dev는 ops를 존중
- Ops는 dev를 존중
- 고정관념 탈피
- 무조건 “no”라고 하지 않기

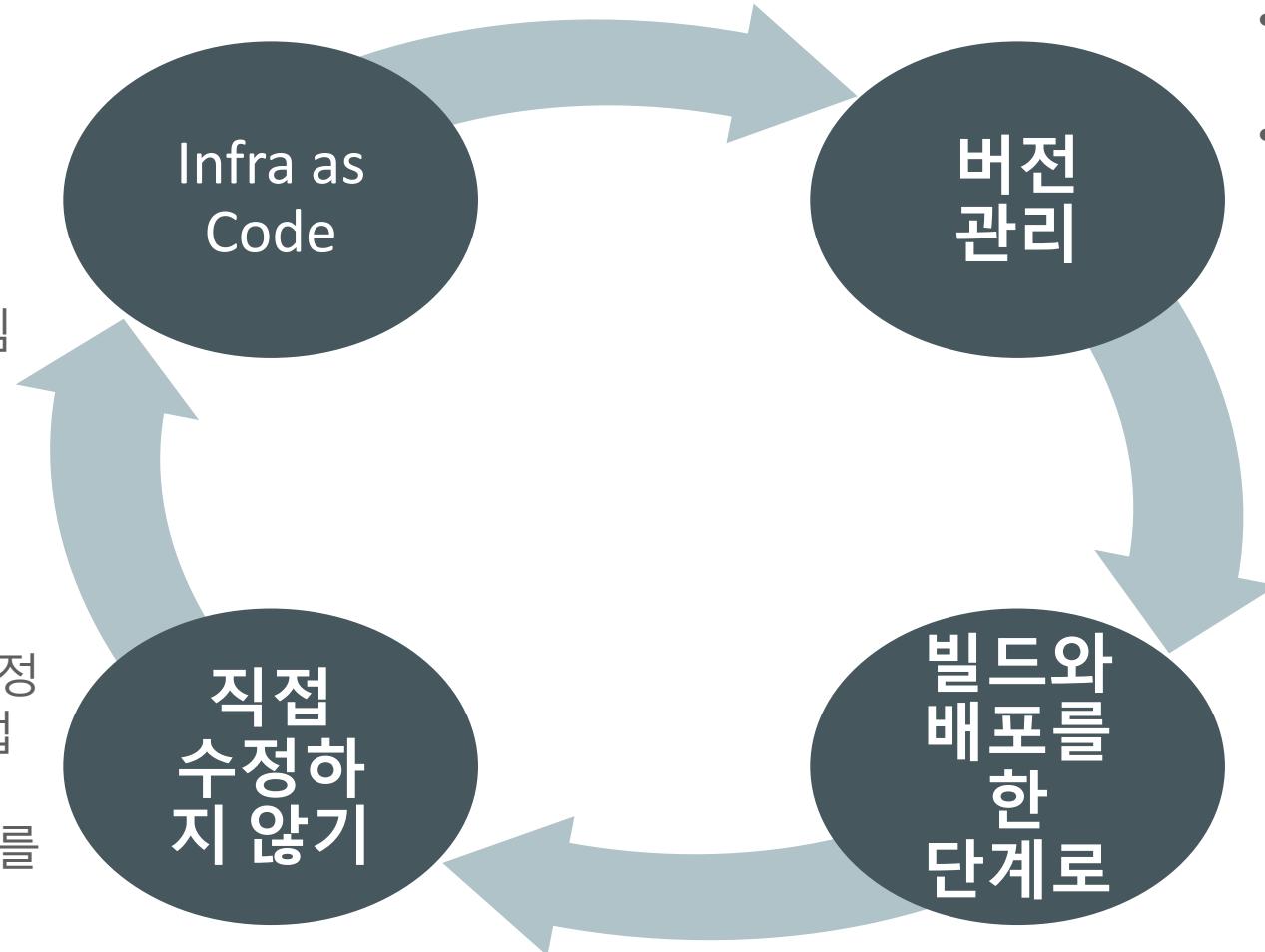


- Dev'의 책임은 코드가 운영될때까지
- 코드를 떠 넘기는 행태는 금물

- 숨지지 않기
 - Dev 토론에 Ops가 참여
 - Ops 토론에 Dev가 참여
 - 운영서/에스컬레이션 계획 공유
 - Dev에게 시스템 접근권한 제공
- No fingerpointing!

DevOps 원칙 #2: 기술

- 환경 구축을 위해 구성관리 툴 사용
- 스크립트는 소스처럼 체크인되고 관리됨
- 오류가 발생하면 수정 후 재배포, 바로 직접 수정하지 말것
- 환경 설정 스크립트를 수정



- 코드 관리를 위해 단일 시스템을 두고 산출물들을 빌드
- 변경이 commit될 때마다 빌드가 트리거 되고 자동적으로 유효성 검증 테스트가 수행됨
- 버튼 하나로 build/deploy (수동)
- build/deploy 스케줄링
- 검증이 실패하면, 중지하고 통보

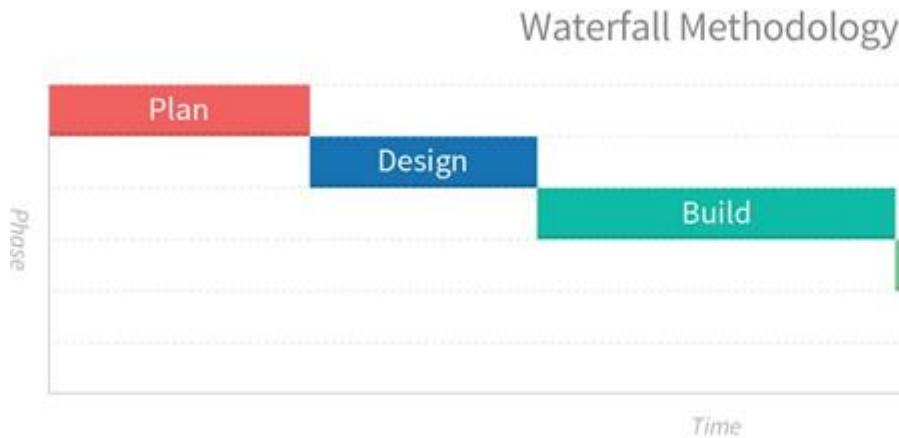
수평 → 수직 구조



작은 수직 구조의 팀에서의 소통은 매우 빠름



Project Management - Agile Methodologies



Pros

- 계획과 설계가 매우 직관적임
- 진척 상황 측정이 쉬움

Cons

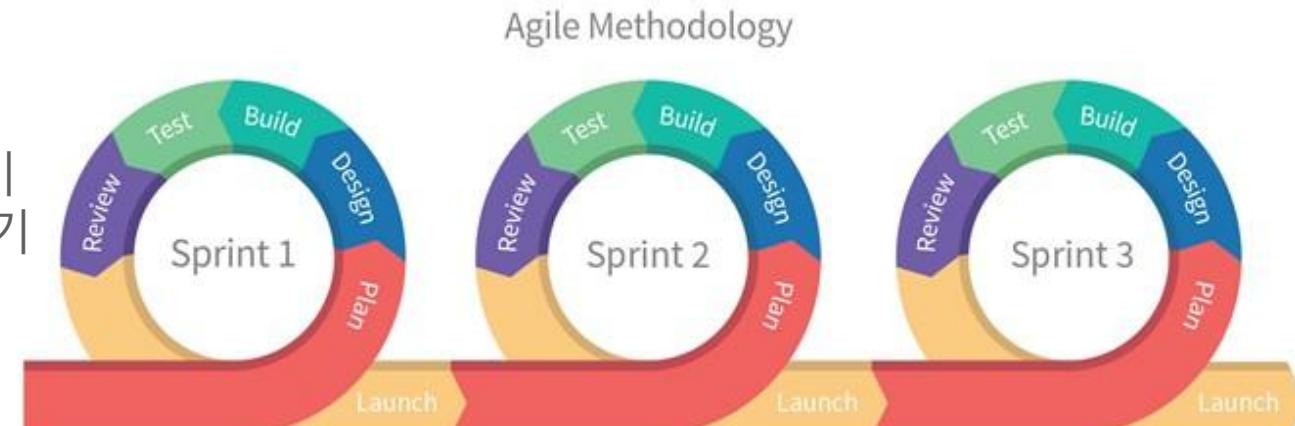
- 고객이 요구사항을 초반부에 잘 형상화 하지 못하는 경우, 고객이 제품 품질에 만족하지 못할 수 있음
- 변경이 어렵거나 많은 비용이 필요할 수 있음

Pros

- 진행되는 상황을 자주 확인할 수 있고, 변경이 용이
- Time to Market이 중요한 애플리케이션의 경우 초기 버전으로 출시가 가능

Cons

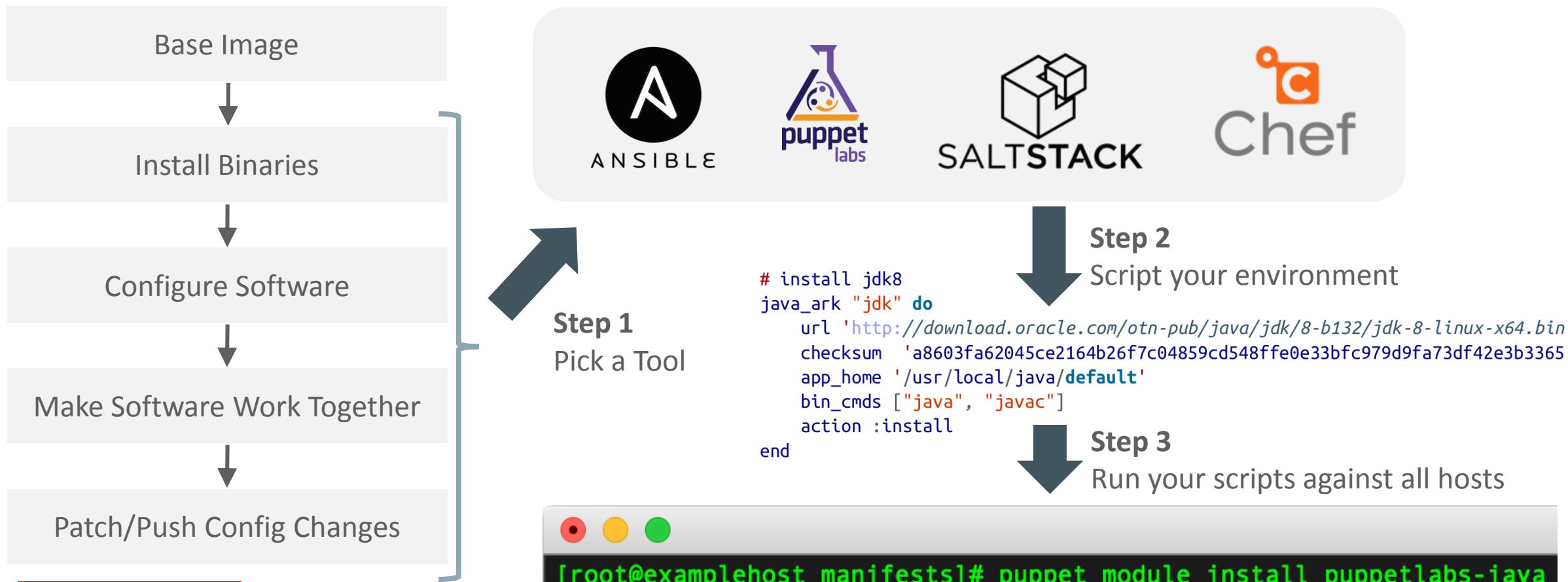
- 고객의 참여가 많이 요구됨
- Time box 기반으로 우선순위 재조정이 계속 일어남



Source : <http://www.screenmedia.co.uk/blog/2014/08/what-is-agile-development-a-brief-introduction/>

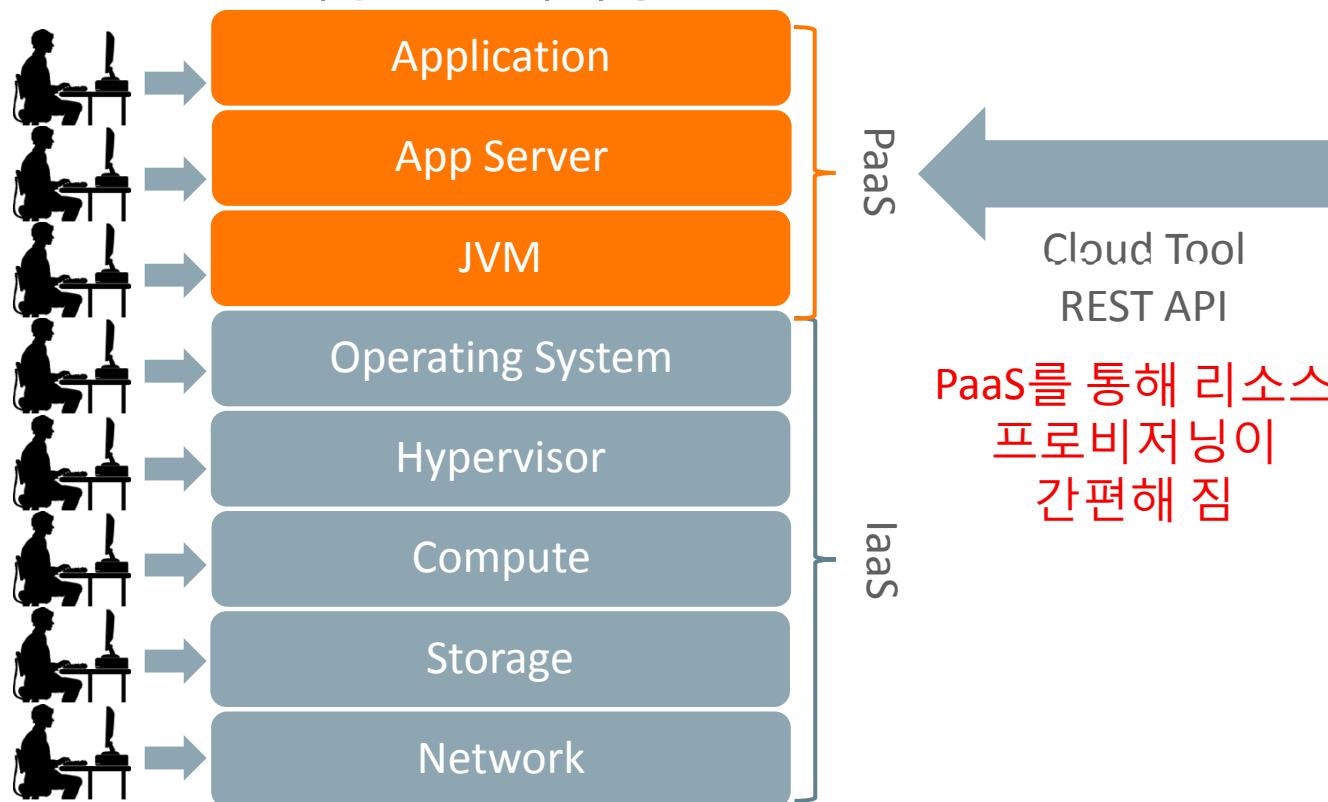
Infrastructure as Code

서버 구성환경을 코드로 저장 관리

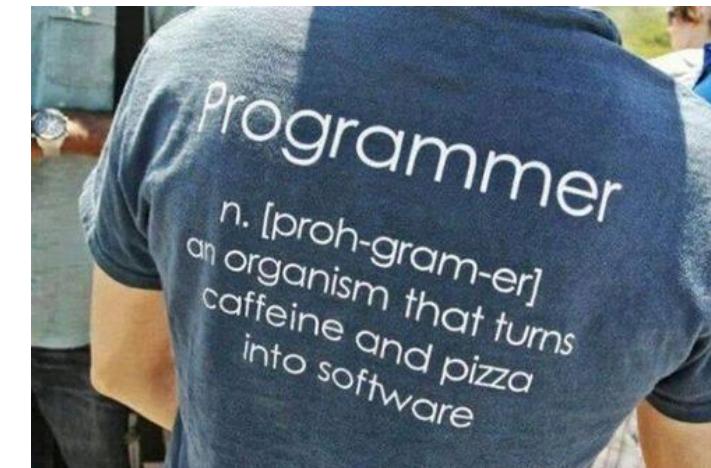


Provisioning - Cloud-like Infrastructure

과거: 운영자(Ops)가 수동으로 각 계층 프로비저닝



현재: 개발자(Dev)가 간편한 툴들을 통해 하드웨어와 소프트웨어 전체 스택을 프로비저닝 할 수 있음



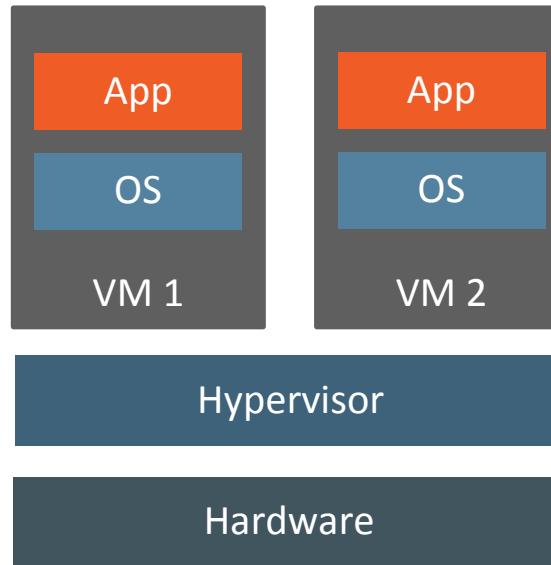
Infrastructure as Code

Provisioning - Lightweight Container

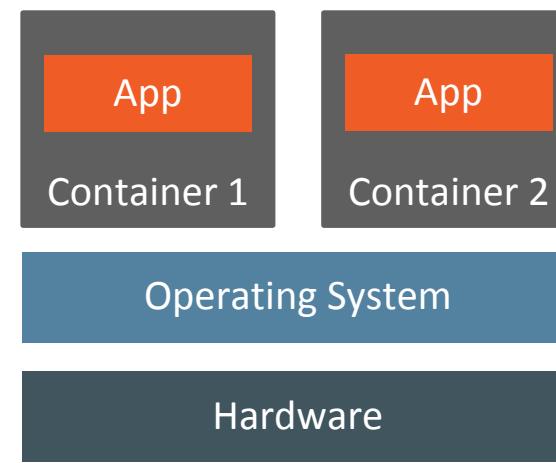
스크립트(Code) 만으로 구동 –
everything is 100% Code.

매우 빠르게 구동 – try for under
10 seconds.

짧끔하게 중지 – ephemeral

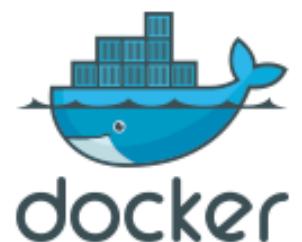


Hardware Virtualization



Containers

Higher density
Easy to start/stop
Portability



표준과 자유

- 필요한 곳은 표준화하고, 나머지는 옵션을 두어 실용적이게 함



Writing Code - 개발 언어는 단지 선택의 문제 Polyglot

Polyglot 환경은 계속적으로 성장

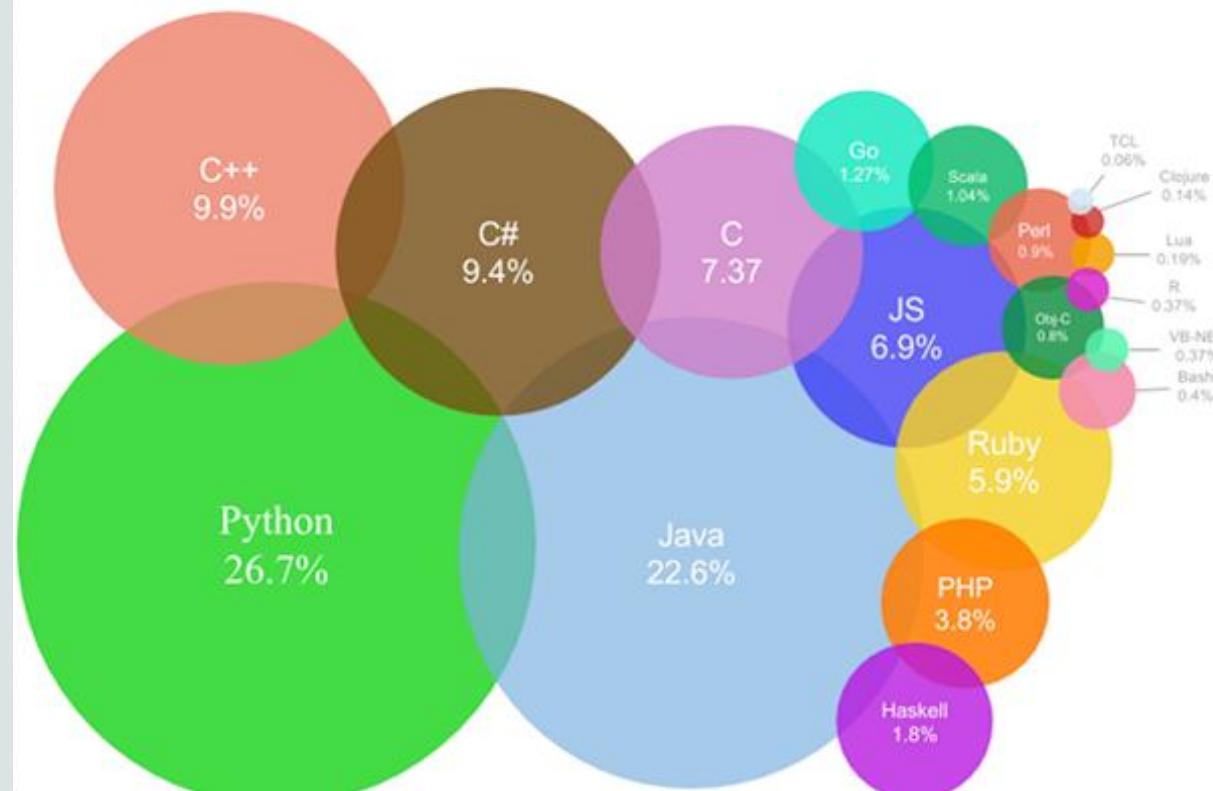


HIBERNATE



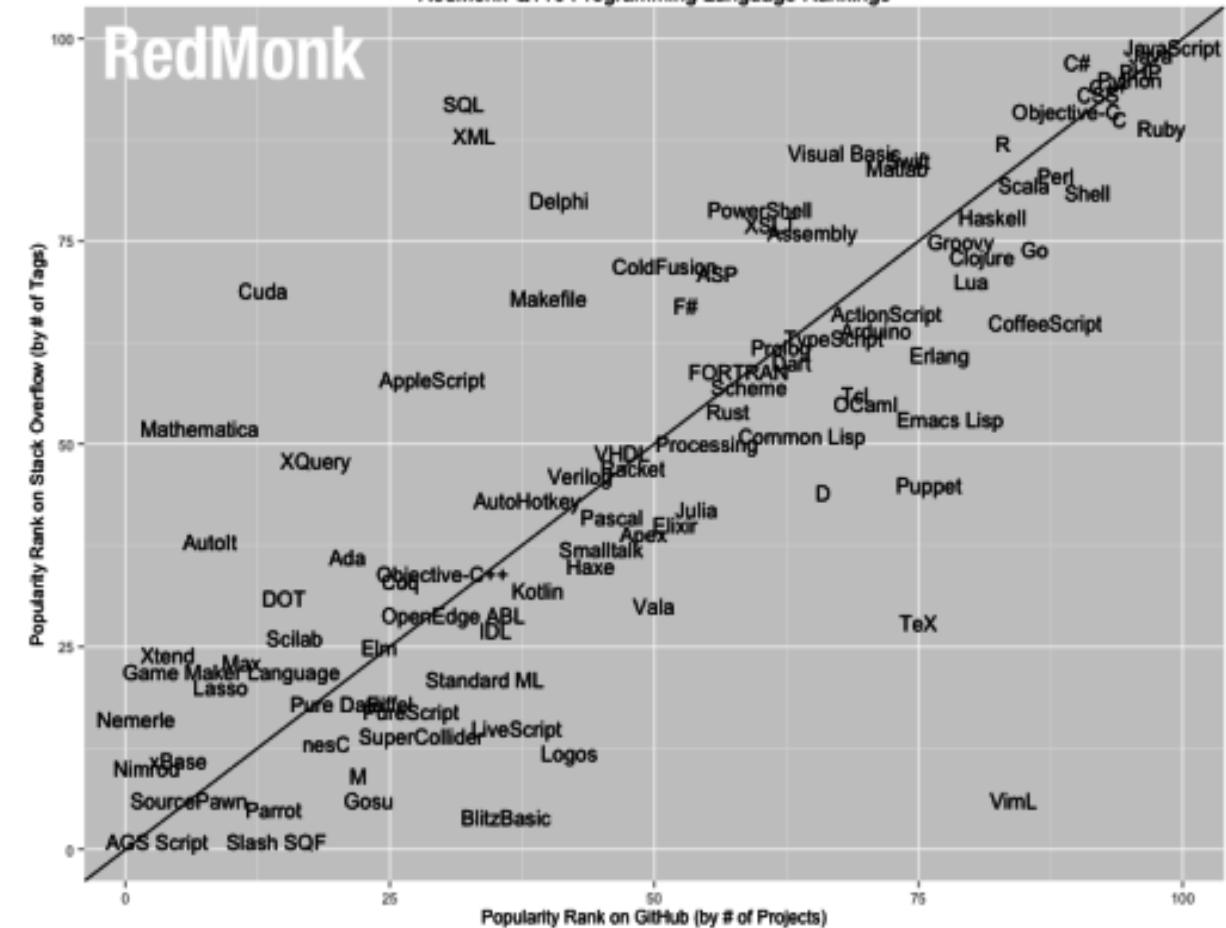
Program Language Trends

Most Popular Coding Languages of 2016

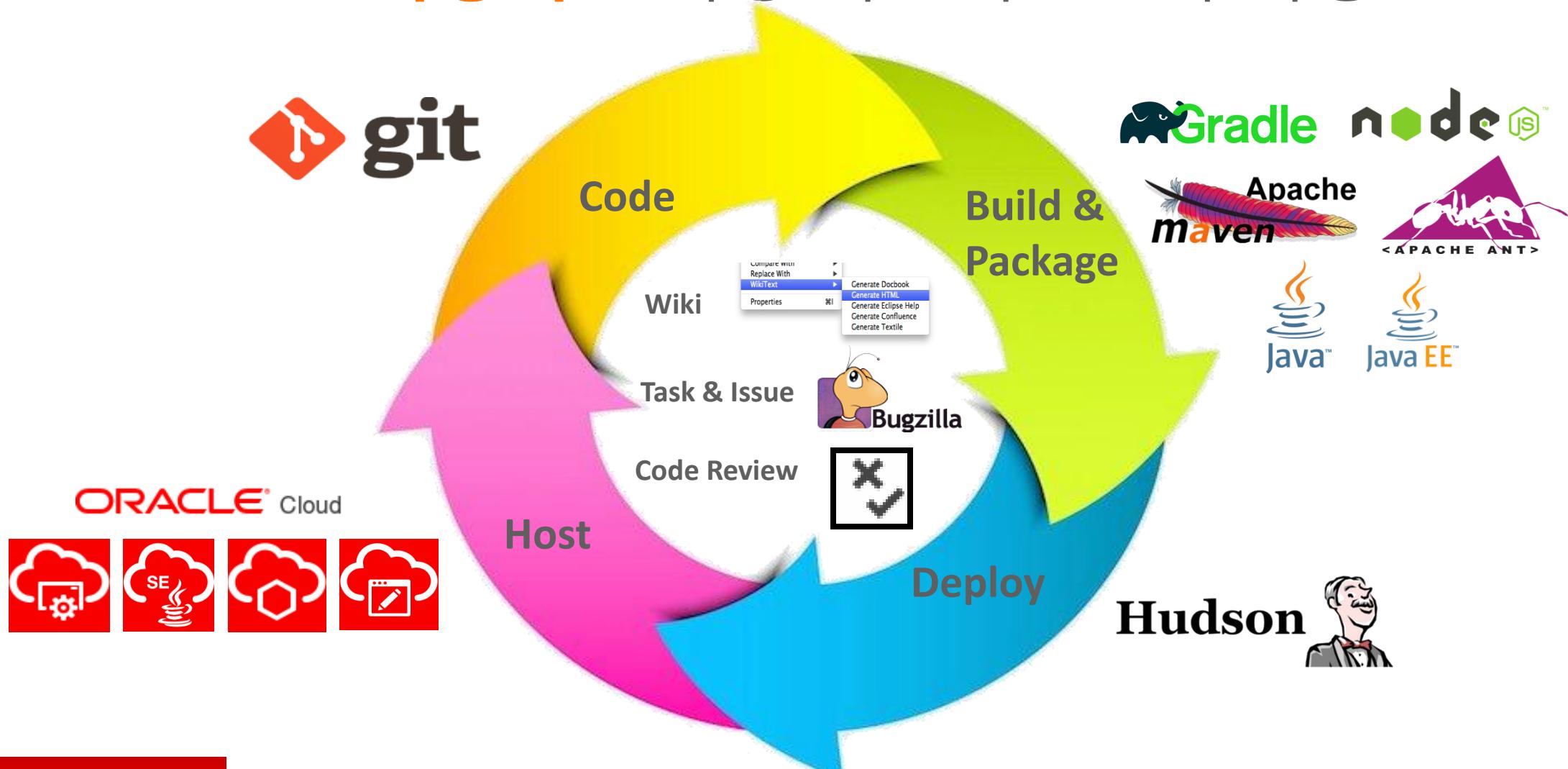


<http://blog.codeeval.com/codeevalblog/2016/2/2/most-popular-coding-languages-of-2016>

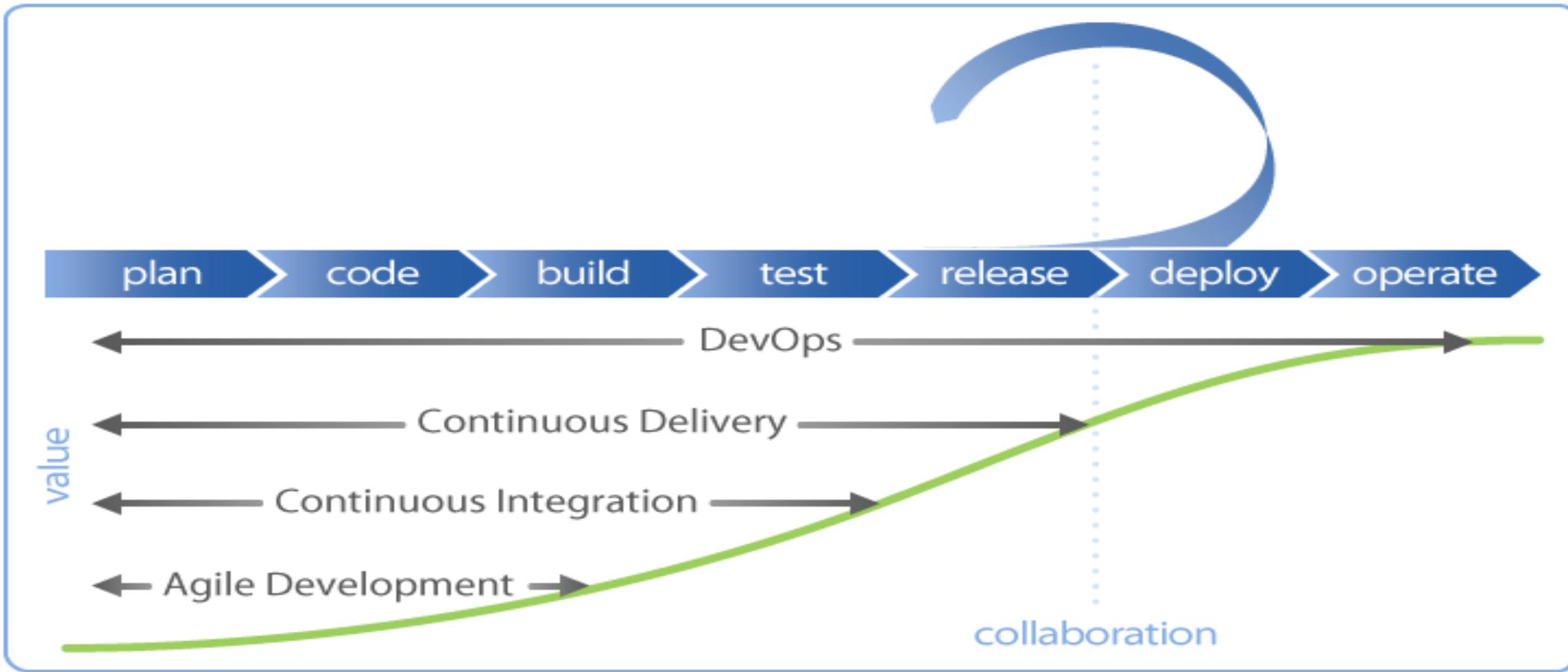
RedMonk Q116 Programming Language Rankings



Automation - 자동화는 가장 기본적인 요구사항



CI, CD, DevOps



<http://www.collab.net/solutions/devops>

많은 좋은 기술들을 활용

Completely Automated By Tools Like These



Capital + mechanical automation always win over labor

Maintaining - 간편한 관리 - One Click

PATCHING BACKUP

Available Patches

As of 28-Jul-2014 3:34:34 AM GMT [Refresh](#)

 JDK Update 1.7.0_70 Readme File	Release Date: 14-Jan-2014 8:00 AM GMT Affected Component: JDK Requires Restart: Yes	Patch Now
--	---	---------------------------

Patch History

 12.1.2.0.2 Readme File	Patched By: weblogic on 23-Jul-2014 12:23 AM G Notes: Need latest patch to keep inst...	Rollback
---	--	--------------------------

PATCHING BACKUP

Incremental backup daily at 6 AM GMT (Full backup every Wed at 6 AM GMT)

As of 28-Jul-2014 3:37:56 AM GMT [Refresh](#)

Last Successful Backup  27-Jul-2014 6:55 AM GMT

[Configure Backups](#) [Back Up Now](#)

Available Backups

 24-Jul-2014 6:55 AM GMT Type: Incremental, scheduled Available Until: 23-Aug-2014 6:55 AM GMT Location: Oracle Storage Cloud Service	Contains:  Notes:	
---	---	---

단 한번의
클릭으로 패치

단 한번의
클릭으로 백업/복구

Maintaining - 간편한 운영 - Easy

Cluster: TestJCS_cluster

Administration Server Domain: TestJCS_domain
Managed Server: TestJCS_server_1
Public IP: 160.34.13.66

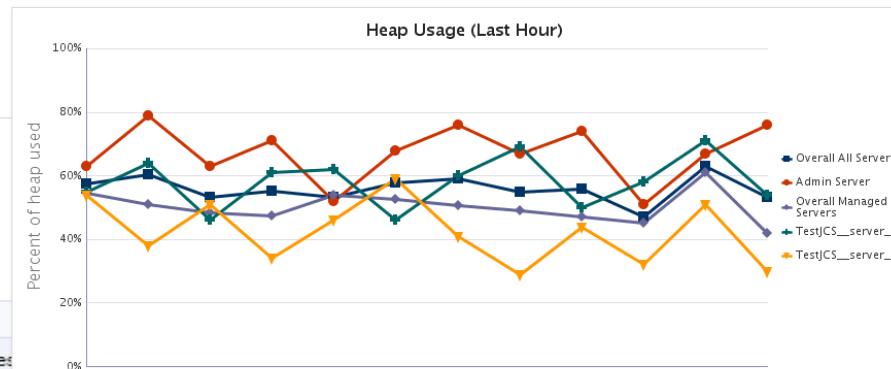
OCPUs: 1
Memory: Restart
Storage: Scale Up/Down

Managed Server: TestJCS_server_2
Host: testjcs-wls-2

OCPUs: 1
Memory: 7.5 GB
Storage: 52 GB

단 한번의 클릭으로
Scale Up/Down
Scale In/Out

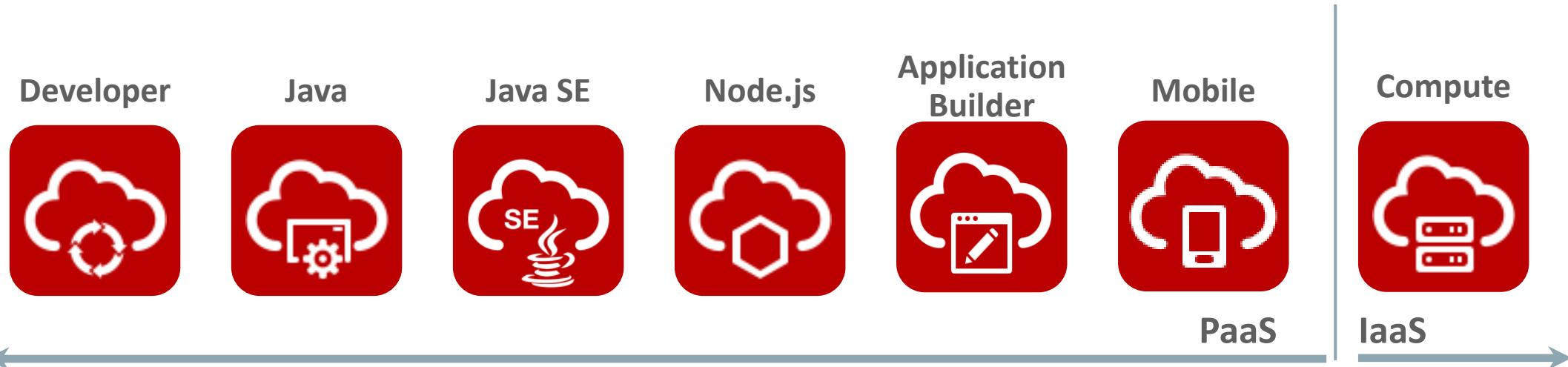
Name	Status	CPU Usage (%)	Heap Usage (MB)	Request Processing Time (ms)	Requests (per minute)		
TestJCS_adm...	▲	0.34	1,346.48	122	0.21	1	0.00
TestJCS_serv...	▲	2.22	1,445.24	0	1.97	0	0.30
TestJCS_serv...	▲	2.51	1,013.65	0	1.97	0	0.30



다양한 모니터링

애플리케이션 개발과 운영을 위한 Oracle Cloud

어떤 비즈니스 애플리케이션이든지 Oracle Cloud로 수용



- 다양한 언어의 애플리케이션 플랫폼 지원:
 - Java: Enterprise Edition(EE), Standard Edition(SE), 경량 프레임워크/컨테이너
 - Polyglot: Node.js, JRuby, Jython, Groovy, Scala, Ruby*, Python*, PHP, Web Profile*
- 공통의 미리 통합된, 팀 개발 환경 제공

* Roadmap Item

Java Cloud: For Java EE Workloads



Key Features

- 모든 기능이 동일하게 제공되는 **WebLogic 11g/12c** 환경
- 클러스터링, In-Memory, 고가용성, 탄력적인 로드밸런싱, Scale Up & Scale Out
- Lifecycle Tooling: 백업/복구, 패칭, 애플리케이션 서버 관리
- Lifecycle Tooling 기능의 사용을 선택할 수 있음
- On-premise와 Cloud 간의 **완벽한 이식성** 제공

Benefits

- 향상된 클라우드 툴을 활용한 셀프서비스 애플리케이션 구축 프레임워크
- 클러스터링을 통한 안전성 및 고가용성 확보
- 완전히 자동화된 프로비저닝, 패칭, 백업 & 복구

Application Container Cloud: For Other Workloads



Key Features

- Node.js 및 Ruby 프레임워크 및 오픈소스를 포함한 어떠한 Java 기반 환경도 지원
- Oracle Cloud 상에 완벽한 Node.js 라이브러리 지원
- IDE 선택 - JDeveloper, Eclipse, NetBeans 및 API access
- Developer Cloud를 연계한 연속적인 개발 환경
- 애플리케이션 라이프사이클 관리를 위한 툴 제공

Benefits

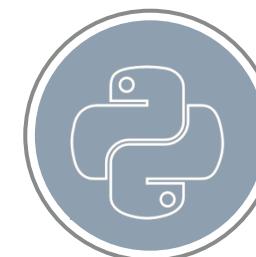
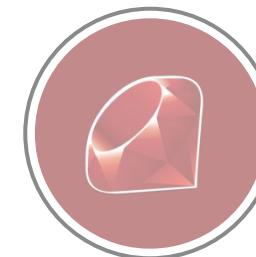
- 향상된 클라우드 툴을 활용한 셀프서비스 애플리케이션 구축 프레임워크
- 완전히 자동화된 프로비저닝, 패칭, 백업 & 복구



Application Container Cloud

클라우드 상에서 non-Java EE 업무를 운영하기 위한 플랫폼

- lightweight microservices를 위한 기반
 - Polyglot: Java SE, Node.js, JRuby, Groovy, Scala, Ruby*, Python*, PHP*, Web Profile*
- 사용이 매우 쉬움
 - 간소화된 최소한의 UI & REST API
- Build, Zip, Deploy
 - 인스턴스 수, 인스턴스 당 메모리 선택
 - 서비스를 위한 Docker 컨테이너를 자동으로 생성
- 백업이 필요 없음 – 애플리케이션은 비상태 유지형
- 자동적인 로드 밸런싱



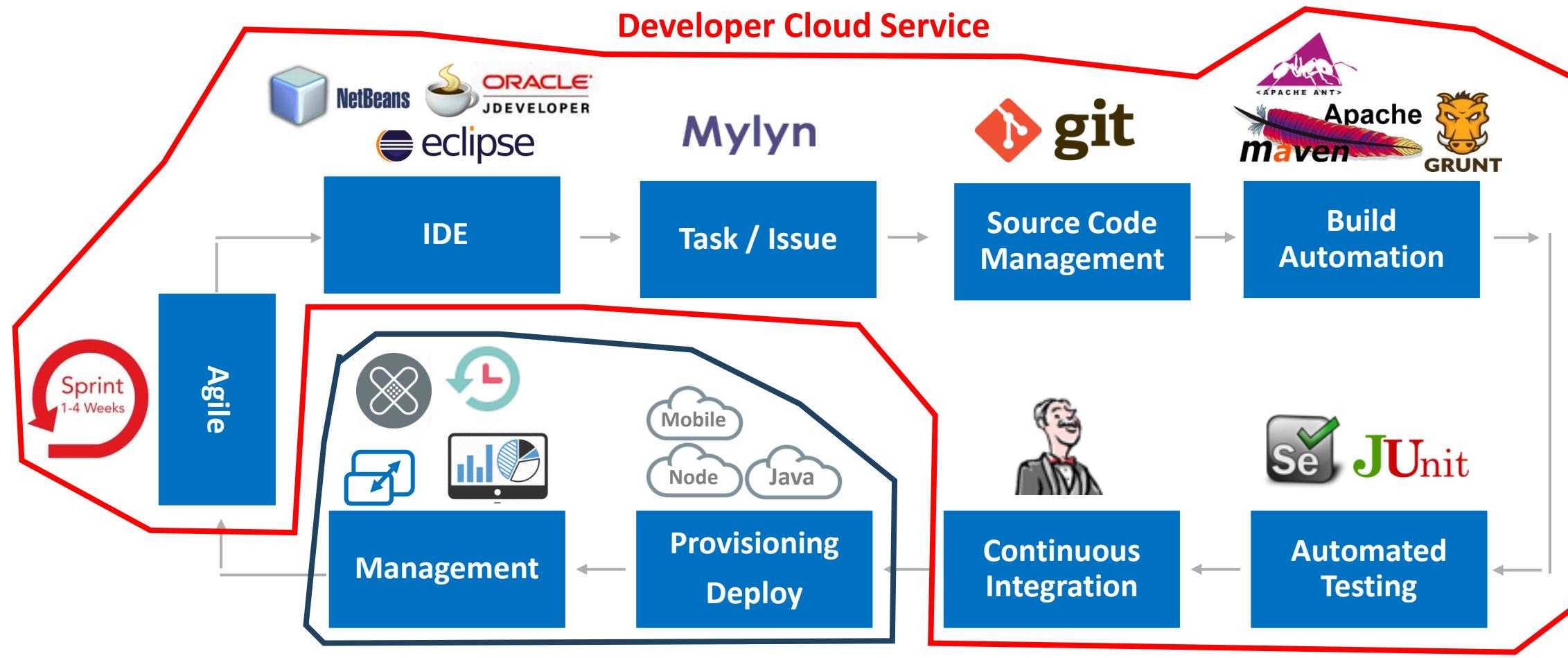
Developer Cloud: For Continuous Delivery



- Java Cloud, Application Container Cloud, Mobile Cloud와 함께라면 무료
- 특징:
 - 프로젝트 기반, 멀티 테넌트
 - wiki server 통합
 - task/defect service 통합
 - 코드 리뷰
 - Git 저장소
 - Maven & Ant 통합
 - Hudson Continuous 통합
 - IDE 통합 (Eclipse, NetBeans, JDeveloper)

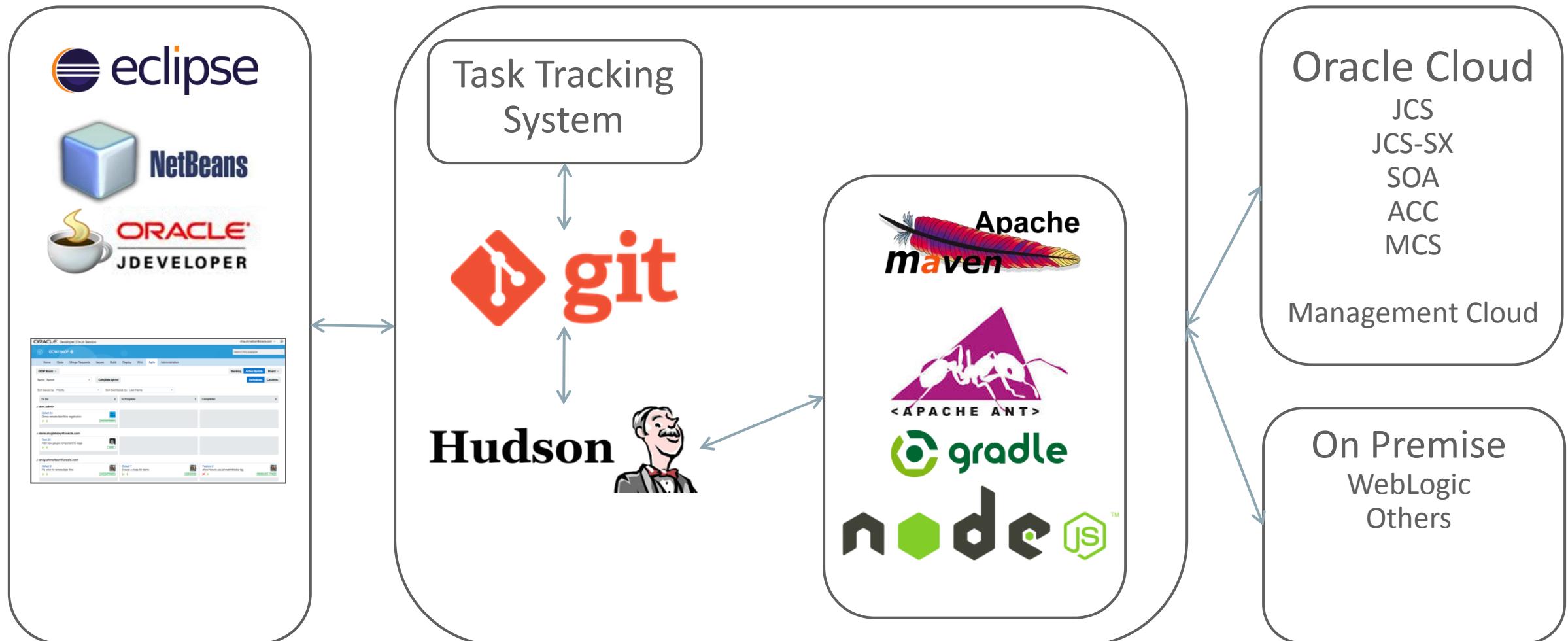


애플리케이션 라이프사이클 자동화



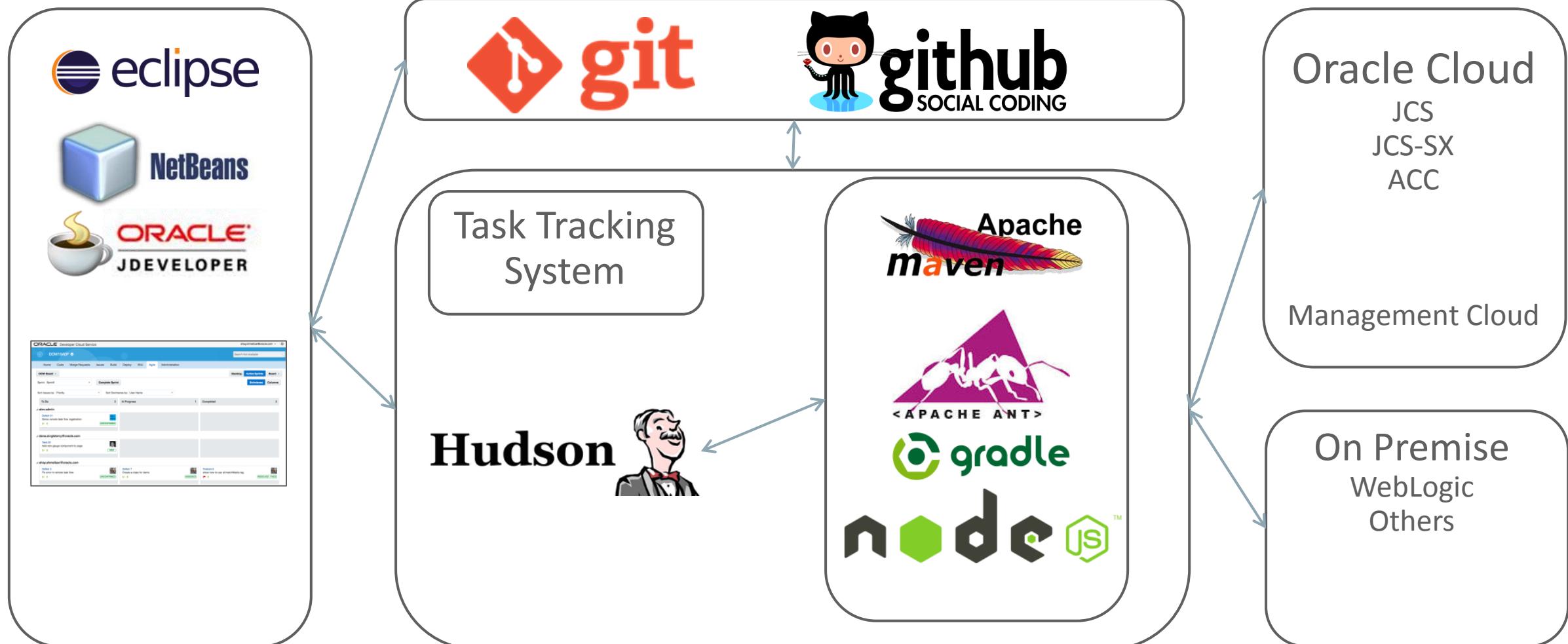


Oracle Developer Cloud Service – Base Architecture



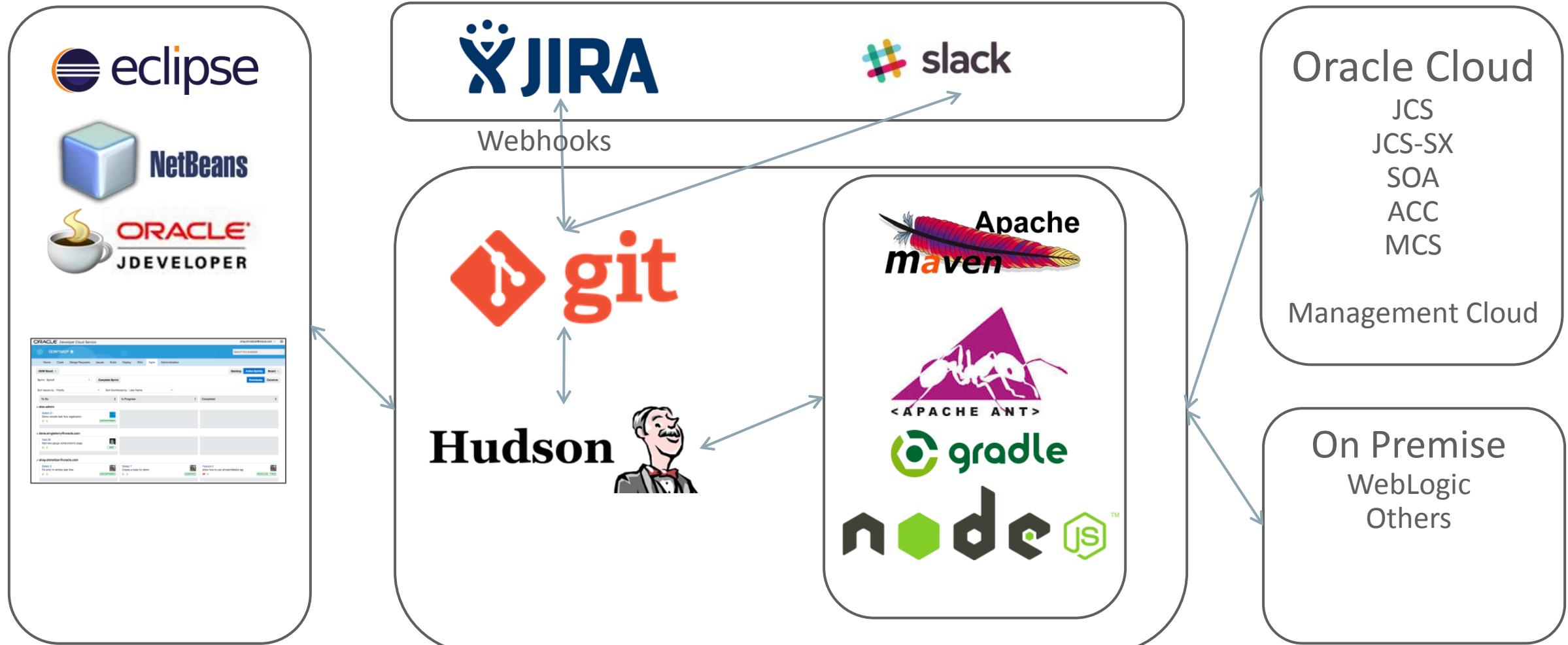


Another Possible Implementation Architecture





Another Possible Implementation Architecture



Oracle Cloud로 빠르게 DevOps 시작

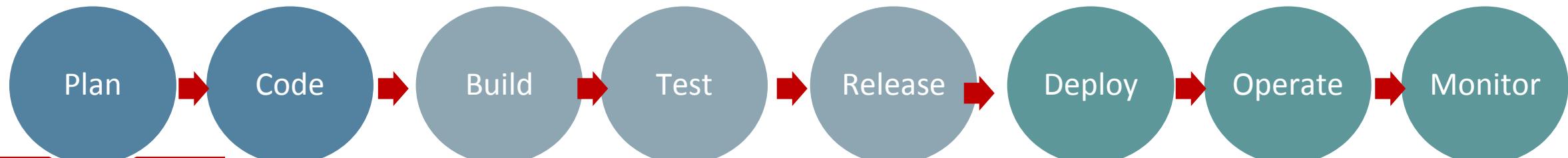
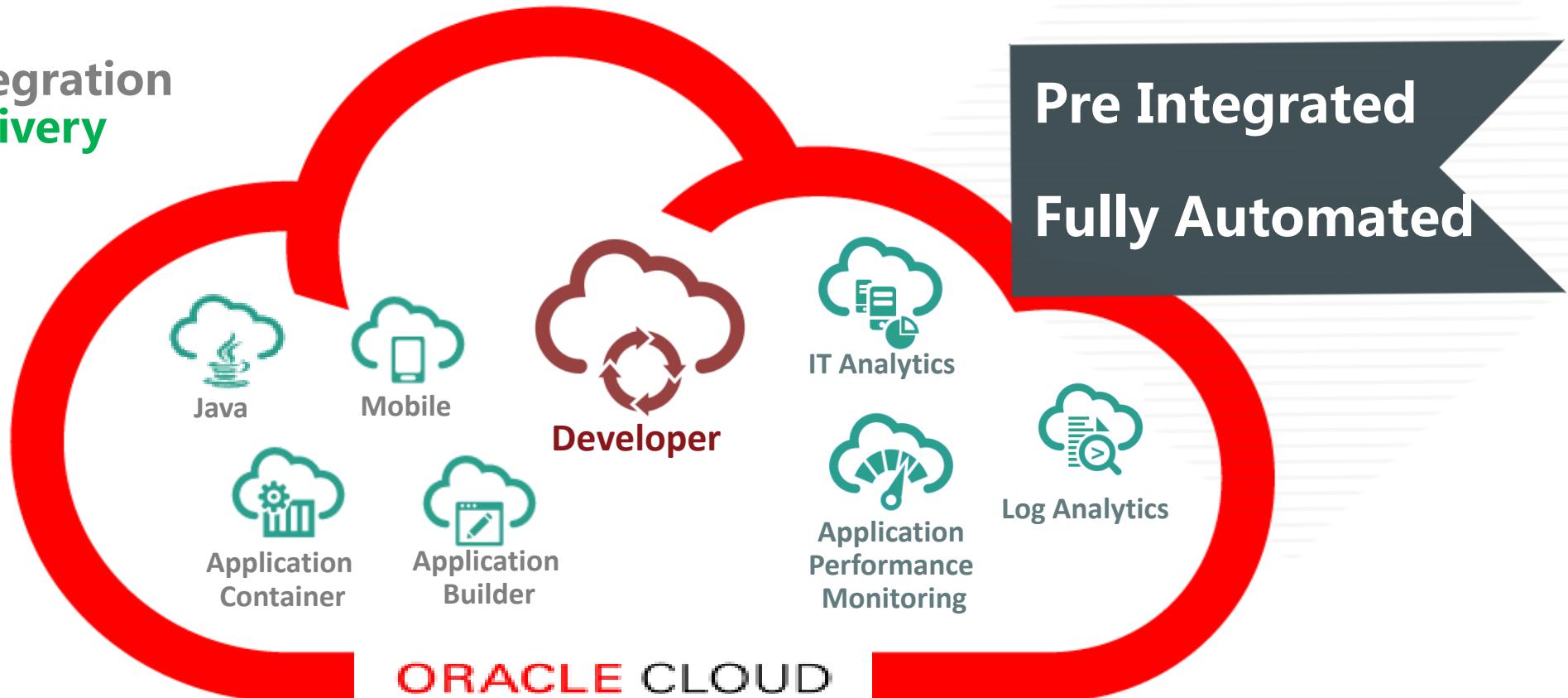
Continuous Integration
Continuous Delivery

Agile
Sprint

Open API

Code Review
Merge

DevOps



ORACLE®

ORACLE®