

# IBM Transformation Advisor / Mono2Micro 서비스를 통한 앱 Modernization 전략

2020. Aug

IBM Korea

## 폭발적인 신규애플리케이션 요구사항과 전망

500m

New digital apps and services by 2023 (IDC)<sup>1</sup>

40%

Technical debt. (Gartner)<sup>2</sup>

<sup>1</sup> Source: IDC FutureScape: Worldwide IT Industry 2020 Predictions, Doc #US45599219, Oct 2019

<sup>2</sup> Source: Gartner: Application Modernization Should Be Business-Centric, Continuous and Multiplatform 15 August 2019, ID G00430084

# IDC estimates ...

**35%**

of new apps produced by means of command line scripting or coding will be **cloud native** by 2022<sup>1</sup>

From **3%**  
to **40%**

**Increase of daily deployments** to production by 2023<sup>1</sup>

**80%**

of apps will be **deployed on containers** by 2024<sup>1</sup>

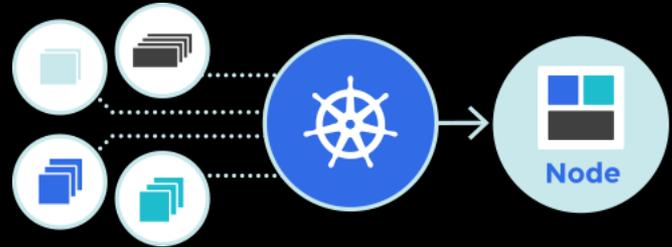
## 컨테이너와 Kubernetes 의 차별화된 가치

**84%** 시장 진입 능력 개선  
**75%** 운영비용 감소 효과  
**60%** IT 운영자의 프로젝트 운영  
능력 개선

Source: Ovum, *Delivering Business Value through  
Transitioning from Managing VMs to Orchestrating Containers*

## 컨테이너 전략의 증대에 따른 관리기법 요구사항 변화

- 컨테이너화된 애플리케이션의 관리 / 배포 자동화
- **Self healing / Auto Scaling**
- 애플리케이션의 자동화된 **Rollout / Rollback**



## Kubernetes 는 컨테이너 **Orchestration** 시장의 진정한 승자

- 컨테이너들의 **77%** 는 Kubernetes 가 관리
- 2017년 이후 **200%** 성장세



Red Hat OCP 는 엔터프라이즈  
Kubernetes 플랫폼 시장의 선도자

최적의  
클라우드  
아키텍처를  
위한 개방형  
표준과  
오픈소스  
지원



안전한 퍼블릭  
클라우드 기반의  
엔터프라이즈급  
소프트웨어

다양하고, 수준 높은  
인더스트리 경험

Linux OS 기반,  
Kubernetes 로  
관리되는 Container  
배포

풍부한 오픈소스  
생태계 지원



기업의 중요한  
워크로드를 다양한  
환경으로 전환 / 실행 /  
관리

# IBM Cloud Pak for Applications



## Optimize: 기존 앱 실행

### WebSphere Application Server

WebSphere ND | WebSphere Base

Liberty Core | Mobile Foundation

### JBoss Enterprise Application Platform



## Unlock : 기존 앱 현대화

### IBM 현대화 및 개발자 도구

모든 구성 요소에 포함

- Transformation Advisor
- Mono2Micro\*
- Application Navigator
- WebSphere Migration Toolkit

로컬 IDE를 위한 엔터프라이즈 개발 도구 및 확장 애플리케이션용 Cloud Pak과 함께 사용할 경우 무료로 지원

Unleash

Unlock

Optimize



## Unleash: 새로운 클라우드 네이티브 앱 구축

### Accelerators for Teams & Enterprise Governance

여러 분야의 팀이 독립된 영역에서 cloud-native 개발. 기업의 보안 및 기술 표준을 준수하며 개발자가 신속하게 개발.

### Red Hat CodeReady Workspaces

Collaborative OpenShift-native IDE

### Enterprise Runtimes

- |                         |                                |
|-------------------------|--------------------------------|
| • Traditional WebSphere | • Node.js                      |
| • Liberty               | • Spring Boot                  |
| • Mobile Foundation     | • JBoss WS                     |
| • Open Liberty          | • Vert.x                       |
| • JBoss EAP             | • Cloud Functions (Serverless) |
| • Quarkus               | • OpenJDK                      |

Distributed Data

SSO

Messaging

## Red Hat OpenShift Container Platform

# IBM Cloud Pak for Applications



## Optimize: 기존 앱 실행

### WebSphere Application Server

WebSphere ND | WebSphere Base

Liberty Core | Mobile Foundation

### JBoss Enterprise Application Platform



## Unlock : 기존 앱 현대화

### IBM 현대화 및 개발자 도구

모든 구성 요소에 포함

- Transformation Advisor
- Mono2Micro\*
- Application Navigator
- WebSphere Migration Toolkit

로컬 IDE를 위한 엔터프라이즈 개발 도구 및 확장 애플리케이션용 Cloud Pak과 함께 사용할 경우 무료로 지원

## Unleash: 새로운 클라우드 네이티브 앱 구축



### Accelerators for Teams & Enterprise Governance

여러 분야의 팀이 독립된 영역에서 cloud-native 개발. 기업의 보안 및 기술 표준을 준수하며 개발자가 신속하게 개발.

### Red Hat CodeReady Workspaces

Collaborative OpenShift-native IDE

### Enterprise Runtimes

- Traditional WebSphere
- Liberty
- Mobile Foundation
- Open Liberty
- JBoss EAP
- Quarkus
- Node.js
- Spring Boot
- JBoss WS
- Vert.x
- Cloud Functions (Serverless)
- OpenJDK

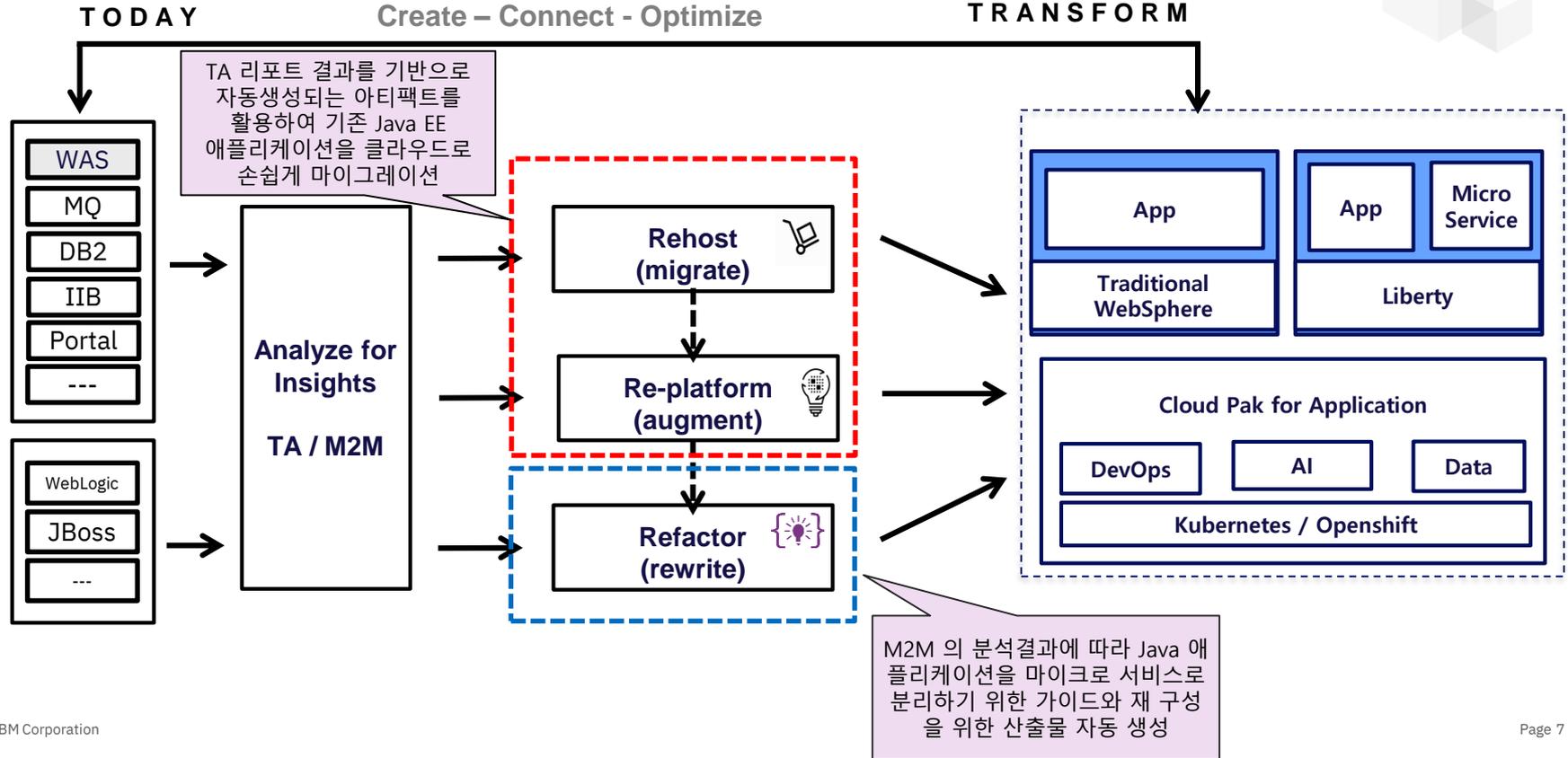
Distributed Data

SSO

Messaging

## Red Hat OpenShift Container Platform

기투자 보호하면서 컨테이너 기반 애플리케이션으로 현대화



# CP4Apps 앱 Modernization 지원 기능 비교



**Unlock:** Modernize and leverage existing investments

	Modernize Operations	Modernize Runtimes	Modernize Architecture
Value			Cloud Native
			Agile Delivery
		Cost Savings	
		Technical Debt Reduction	
		Flexible Deployment	
CP4Apps Recommended End State	Application	Application	Microservices
	tWAS	Liberty	Liberty
	Containers	Containers	Containers
	OpenShift	OpenShift	OpenShift
Mod Tools	Transformation Advisor		Mono2Micro (beta)

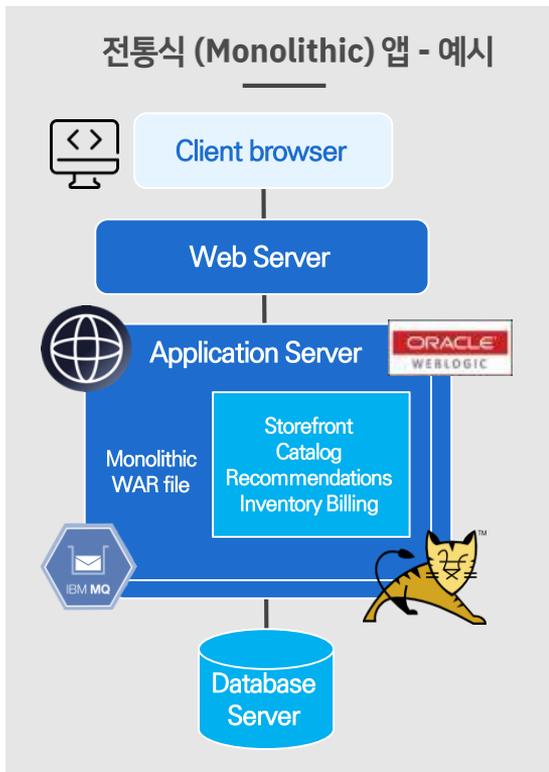
■ New  
■ Existing



# IBM CloudPak for Applications – Modernization Toolsets

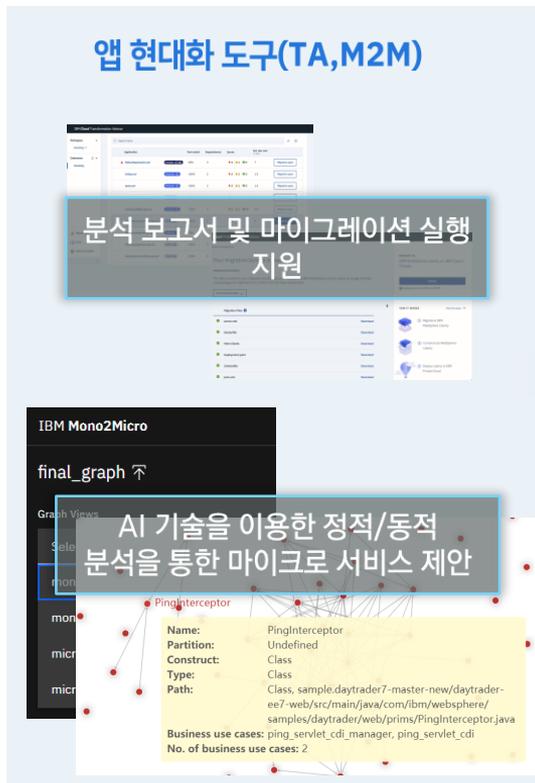
기존 애플리케이션을 현대화하기 위한 분석과 마이그레이션, 마이크로 서비스로의 전환을 가속화 하고, AI 기술을 활용

## Modernization & Developer Tools



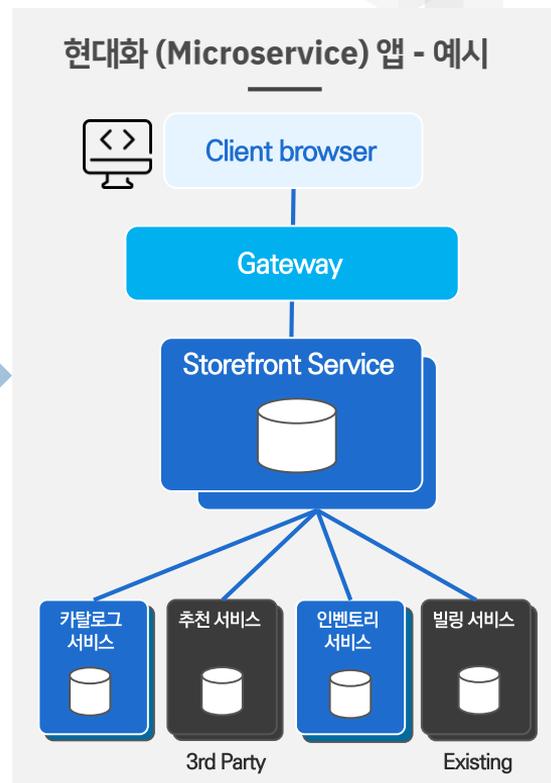
분석

평가



계획

실행





## 마이그레이션 분석 및 예측

APPLICATION	RECOMMENDATION	COMPLET	TECH MATC	DEPENDENCIES	ISSUES	DEV	DEPLOY	TOTAL	
myapp@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details
libr@libr	Library on Private Cloud Server Environment or Cloud Location of	Done	Done	Done	0	0	0	0	Migration Details

## 상세 분석 리포트



## 인벤토리 리포트 제공

The image shows a 'Detailed Migration Analysis Report' with a summary of rule severity and connectivity rules. It includes counts for rules flagged, total results, and a table of connectivity rules.

SYMBOL	LABEL	RULES FLAGGED	TOTAL RESULTS	DESCRIPTION
🔴	Severe	2	8	Severe rules include an API removal or behavior change that can break the application and must be addressed.
🟡	Warning	7	85	Warning rules indicate behavior changes that might break the application and should be addressed.

## 마이그레이션 수행 지원

The image shows a 'Your migration bundle is ready' page. It includes a 'Migration Flow' section with steps like 'Inventory', 'Analyze', 'Plan', and 'Execute'. Below that is an 'Application Dependencies' section with a table of dependencies.

IBM Transformation Advisor 를 통해 고객의 애플리케이션을 신속하게 마이그레이션 지원

컨테이너 관리영역

표준기반 이식성 확보

가용성 극대화

애플리케이션 확장성 확보

멀티 클라우드 전환

개발자 생산성 확보



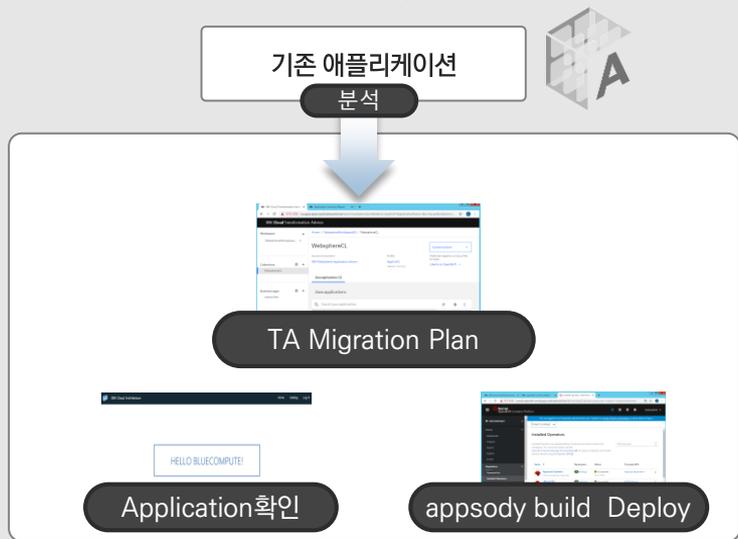
# IBM Cloud Transformation Advisor를 통한 마이그레이션

기존 WAS 에서 IBM Cloud Transformation Advisor 툴을 활용하여 애플리케이션 현황 분석 후 OpenLiberty로 마이그레이션 빌드 배포 예시

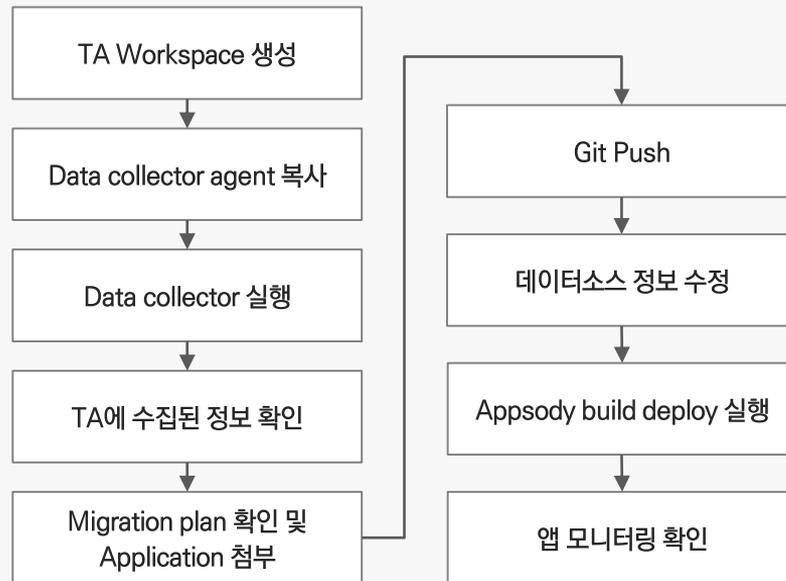
## Transformation Advisor 를 통한 마이그레이션 지원 기능 점검

### Migration

As-Is WAS에서 서비스 중인 애플리케이션을 TA 분석 후 Appsoy 빌드 배포



### 마이그레이션 절차



# Mono 2 Micro (M2M) overview



Unlock: Modernize and leverage existing investments

## 애플리케이션 전환 문제점

- 기존 코드를 분석할 수 있는 수준 높은 이해와 분석능력 필요
- 작업을 반복적으로 수행해야 하고, 시간이 많이 소요됨
- 아키텍처의 재구성이나 재구현을 필요로 함

No Simple  
Non Disruptive  
No Economical

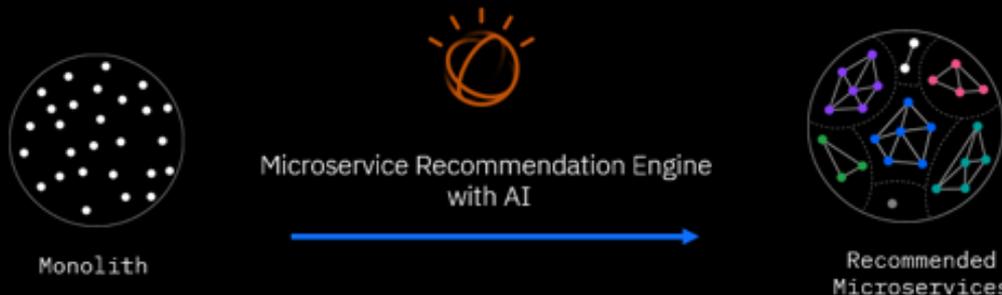


Full-scale 애플리케이션 전환  
클라우드 활용 아키텍처 필요

## IBM 제안 솔루션

# Mono2Micro

AI based automatic transformation of monoliths into microservices



- 아키텍처가 리팩토링을 수행하기 위한 다양한 기능을 통해 수작업에 따른 각종 부작용과 시간, 비용을 절감
- 자동화된 마이크로 서비스 권고 사항 이행
- 데드코드를 식별하여 마이크로 서비스 이행을 가속화



# IBM Cloud Pak for Applications – 기존 앱 현대화 Mono2Micro

## 소스코드 정적분석



```
Output >>> /var/application/sample.daytrader7-klu
drwxr-xr-x. 11 root root 4096 Aug 2 22:40 sample.daytrader7
drwxr-xr-x. 11 root root 4096 Aug 2 23:10 sample.daytrader7-klu
```

```
...
    * Constructor for CompletedOrdersAlertFilter
    public OrdersAlertFilter() {
        super();
    }
    System.out.println(String.valueOf(System.currentTimeMillis())+"["+String.
    valueOf(Thread.currentThread().getId())+"]:"+"Executing
    sample.daytrader7/daytrader-ee7-web/src/main/java/com/ibm/websphere/sample/
    daytrader/web/OrdersAlertFilter.java:class:OrdersAlertFilter;OrdersAlertFi
    lter?);
    System.out.println(String.valueOf(System.currentTimeMillis())+"["+String.
    valueOf(Thread.currentThread().getId())+"]:"+"Executing
    sample.daytrader7/daytrader-ee7-web/src/main/java/com/ibm/websphere/sample/
    daytrader/web/OrdersAlertFilter.java:class:OrdersAlertFilter;OrdersAlertFi
    lter?);
}
```

## 다양한 리포트 제공

IBM Mono2Micro

final\_graph ↑

Graph Views

Select a view

- mono\_run\_time\_c
- mono\_data\_deper
- micro\_detail\_part
- micro\_detail\_part

Name: PingInterceptor  
Partition: Undefined  
Construct: Class  
Type: Class  
Path: Class, sample.daytrader7-master-new/daytrader-ee7-web/src/main/java/com/ibm/websphere/samples/daytrader/web/prims/PingInterceptor.java  
Business use cases: ping\_servlet\_cdi\_manager, ping\_servlet\_cdi  
No. of business use cases: 2

## 애플리케이션 동적분석

- daytrader-trace
  - contexts
    - daytrader.json
  - logs
    - console.log
  - tables
    - refTable.json
    - symTable.json

## Microservice 전환지원

cardinal	2020-07-29 오후 3:41
oriole	2020-07-29 오후 3:22
sample.daytrader7	2020-07-29 오후 3:40
sample.daytrader7-partition_0	2020-07-29 오후 3:41
sample.daytrader7-partition_1	2020-07-29 오후 3:41
sample.daytrader7-partition_2	2020-07-29 오후 3:41
sample.daytrader7-partition_3	2020-07-29 오후 3:41
sample.daytrader7-root_dest_undefined_us...	2020-07-29 오후 3:41
sample.daytrader7-src_dest_undefined_use...	2020-07-29 오후 3:41
Cardinal-Report.html	2020-07-29 오후 3:22
coverage.txt	2020-07-29 오후 3:21
graph_view.txt	2020-07-29 오후 3:22
Oriole-Report.html	2020-07-29 오후 3:22

IBM Mono2Micro 는 전통적인 고객의 애플리케이션을 자동으로 분석하고, Microservice 화 지원

데이터와 앱 정적분석

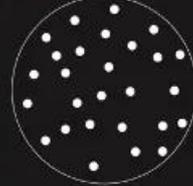
표준 Java 앱 지원

테스트케이스별 동적분석

다양한 관점의 리포트

마이크로 서비스 전환가이드

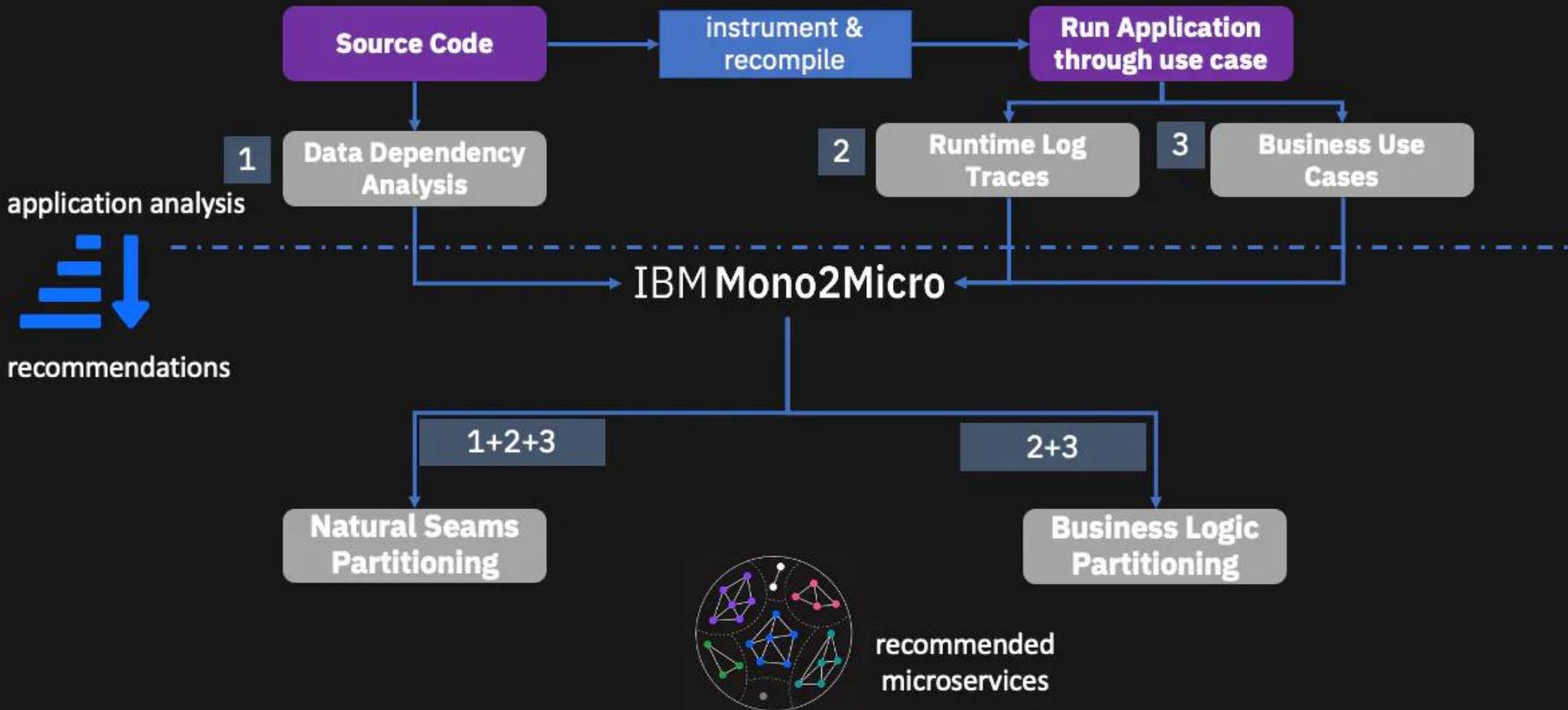
# Mono2Micro Workflow



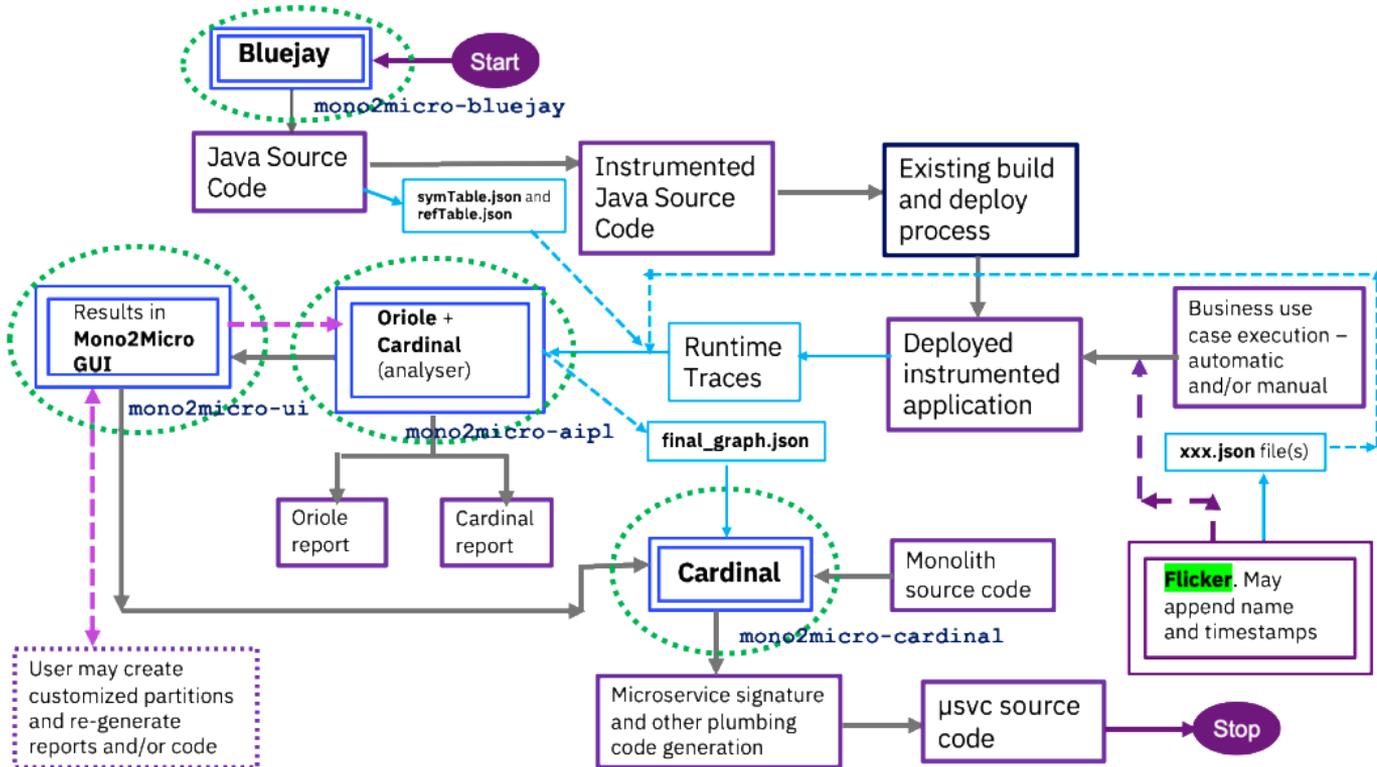
application  
monolith



Unlock: Modernize and leverage  
existing investments



# Mono2Micro Components & Workflow



Bluejay : 자바코드를 정적 분석 하고, 동적분석을 위한 코드주입

Flicker : 테스트 케이스별 분석 보조 프로그램

Oriole : Mono2Micro 의 AI 엔진

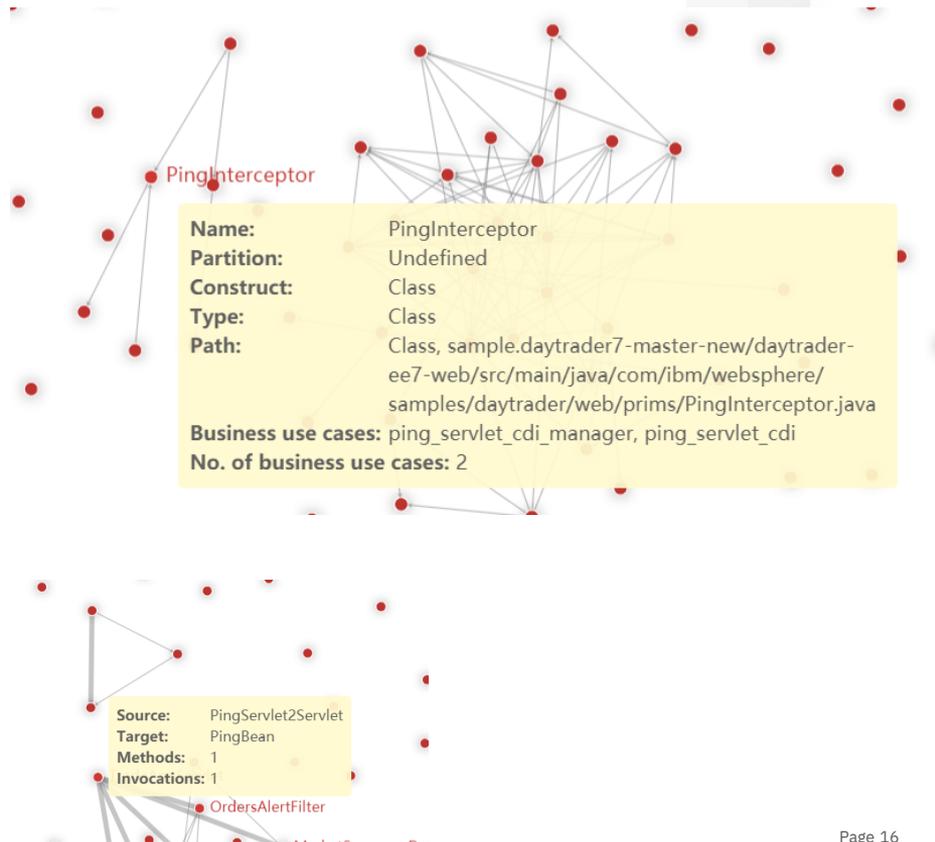
Cardinal : 각종 리포트 및 코드 주요 부분 생성 지원



# 애플리케이션이 실행시 호출 상관 관계를 도식화

## Runtime Call Graph

- 런타임 콜 그래프 (Runtime Call Graph)
  - ✓ 모놀리식 애플리케이션내 클래스들이 실행되면서 호출하는 (되는) 관계 및 상세 내역을 그래프로 표현
- 일반적으로 런타임 호출 그래프 에서 다음과 같은 경우에 해당하는 격리된 (Isolated) 여러 클래스를 식별
  - ✓ 실행된 테스트 케이스의 커버리지 부족
  - ✓ 단순하게 실행되지 않는 DeadCode 인 경우
  - ✓ 1회성으로 초기화 코드만 가지고 있는 경우

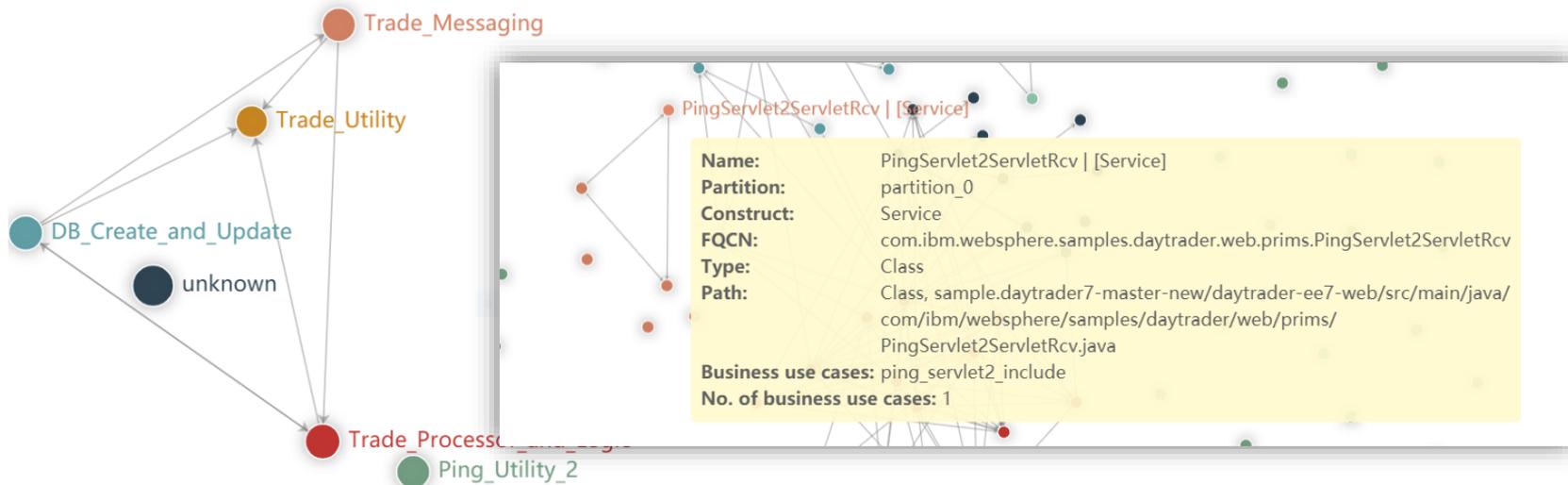




# 비즈니스 로직에 기반한 그룹화를 통해 권장되는 마이크로 서비스를 식별

## Microservices 권고 사항

- **Mono2Micro** 는 비즈니스 로직이 수행되면서 발생하는 트레이스의 컨텍스트를 통해 클래스들 사이에 어떠한 순서로, 어느 정도의 빈도수로 상호 작용하는지 분석 (**Business Logic recommendations**)
- **AI** 기술에 기반한 비즈니스 로직 기반 클래스 생성, 런타임 호출 추적의 그룹화를 캡처
- 인과관계 / 사용사례, 유사성 / 기타 시간적 관계 / 클래스 자체와 클래스의 행동양식

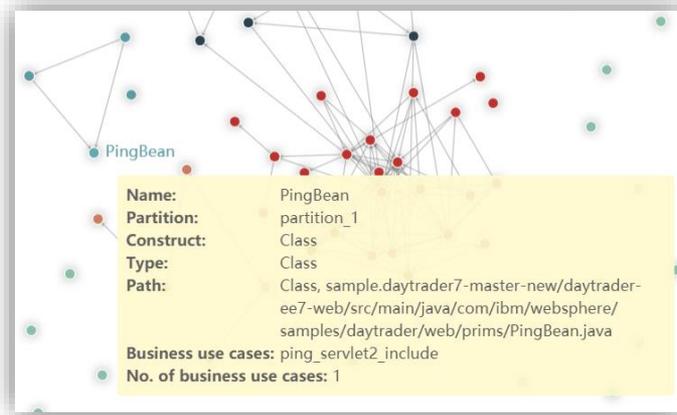
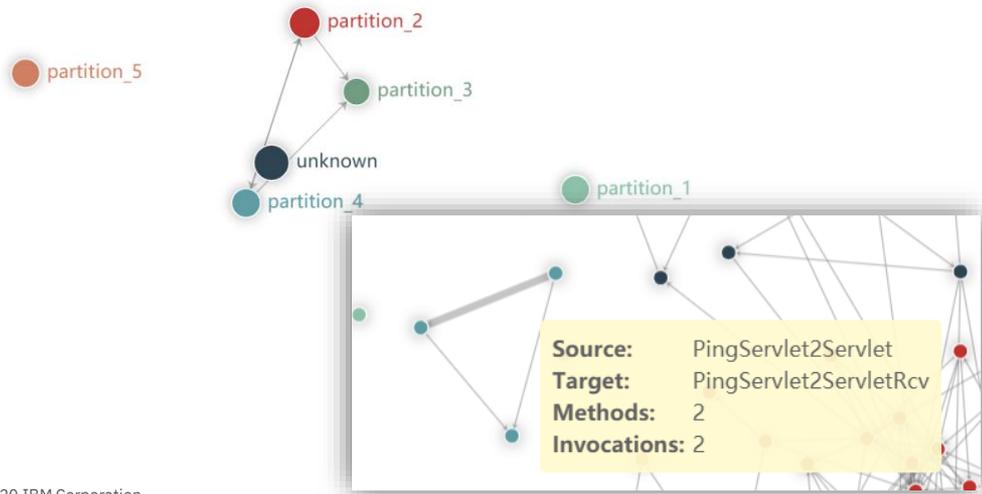




# 데이터 연관관계를 고려한 Natural Seams 마이크로 서비스 식별

## Microservices 권고 사항

- **Mono2Micro** 는 데이터 종속성에 대한 분석을 통해 비즈니스 로직 기반 그룹화를 더욱 강화
- 관련 클래스 그룹을 데이터 종속성과 반복적으로 병합하여 **Natural -seams** 기반 그룹을 생성
- 이러한 그룹화를 통해서 **Mono2Micro** 는 기존 클래스들을 다시 작성해야 할 필요성을 최소화 시킴





## Microservice 권고 가이드 지원 – Meet in the middle

새로운 마이크로 서비스 파티션을 생성하거나, 기존 클래스들을 파티션으로 이동하도록 하여 커스터마이징 된 마이크로 서비스 파티션 제안을 제작 하기 위해 이용되는 기법 지원

PingServlet2ServletRcv

Move a class to different partition

Source partition:  
partition\_1

Target partition:

Chose destination partition

- partition\_2
- partition\_3
- partition\_4
- unknown

Cancel Ok

Create New Partition

The partition 'partion\_x' does not exist, would you like to create it?

Cancel OK



# M/L, D/L 기반 파티션 생성을 지원하기 위한 기타 리포트

## 각종 Summary Report

- 해당 파티션 내에 존재 하는 클래스들의 숫자
- 특정 파티션에서 사용되는 비즈니스 유스케이스의 이름과 빈도수
- 파티션 들 사이의 호출관계에 대한 리포트 생성
- 모든 호출에 따라 파티션들로 전달된 모든 **Non-Primitive** 매개변수와 파티션 내에서 참조하는 외부 클래스
- 마이크로 서비스로 구현시 **REST** 호출에 필요한 **Signature** 를 정의하고, 결정하기 위한 인터페이스

Partition Name	Classes	Cases	Business Use Cases
Trade_UTILITY	2	16	<ul style="list-style-type: none"> <li>• account</li> <li>• config</li> <li>• int</li> <li>• login</li> <li>• logout</li> <li>• market_summary</li> <li>• market_summary_dictionary</li> <li>• ping_server_cell</li> <li>• ping_server_cell_manager</li> <li>• ping_server_writer</li> <li>• primitives</li> <li>• recreate_db_tables</li> <li>• register</li> <li>• repopulate_db_tables</li> <li>• trade_buy</li> <li>• trade_sell</li> </ul>
Ping_UTILITY_1	6	2	<ul style="list-style-type: none"> <li>• ping_server_cell</li> <li>• ping_server_cell_manager</li> </ul>

**Mono2Micro Business Use Case Partition Analysis - Daytrader7**

Summary

Business Use Cases per Partition

DB\_Create\_and\_Update

Member Class Details ( 9 )

Ping\_UTILITY\_1

Member Class Details ( 6 )

External-Facing Classes (Other Partitions Invoke These Methods)

Class Name

AccountDataBean [Persistence] implements Serializable

AccountProfileDataBean [Persistence] implements java.io.Serializable

MarketSummaryDataBean implements Serializable

OrderDataBean [Persistence] implements Serializable

Methods (Invoked by Other Partitions)

AccountDataBean [overloaded\_#002]

AccountProfileDataBean [overloaded\_#002]

MarketSummaryDataBean [overloaded\_#002]

OrderDataBean [overloaded\_#002]

**Mono2Micro DEEP PARTITION ANALYSIS - Daytrader7**

Trade\_Processor\_and\_Logic Partition

Member Classes (12)

External-Facing Classes (Other Partitions Invoke These Methods)

Non-Primitive Object Parameters Passed From Outside To This Partition

Partition Invokes the Following Class.Method(s) Residing Outside This Partition

Partition Has References To The Following Classes Outside The Partition

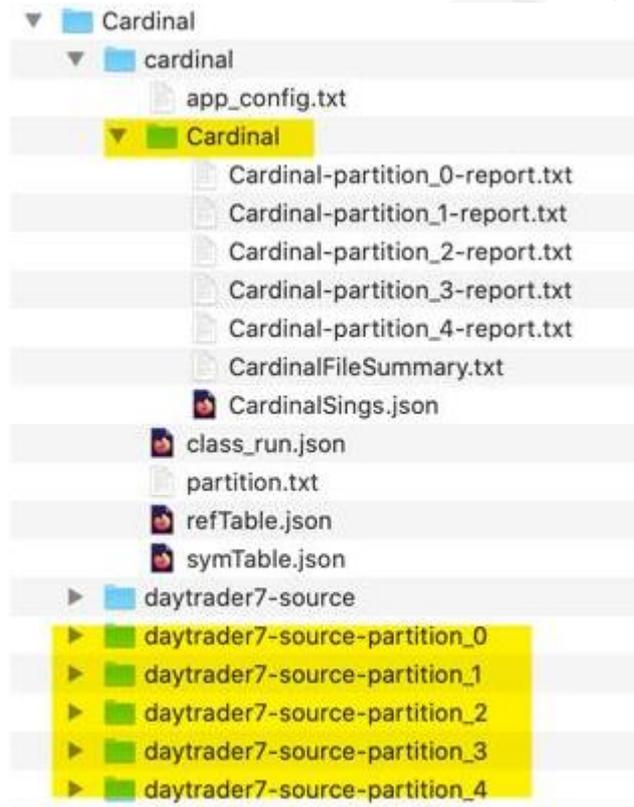


## Microservice 권고 가이드 지원 – Generating Code

분석된 결과에 따라 클래스들을 마이크로 서비스 파티션별로 분리하고, 보조 클래스들을 자동으로 생성



COMPLETED  
Report Output >>> /var/cardinal/cardinal/Cardinal



# IBM Cloud Pak for Applications

## Application Modernization

기 개발된 Application을 클라우드로 전환할지, 한다면 어떻게 어떤 방식으로 클라우드로 전환할지 고민

- Re-Host, Re-Platform, Re-Factor

## Cloud Native

컨테이너 개발 기술에 익숙하지 않고, Microservice Architecture APP, Cloud Native APP은 어떻게 개발할지 고민

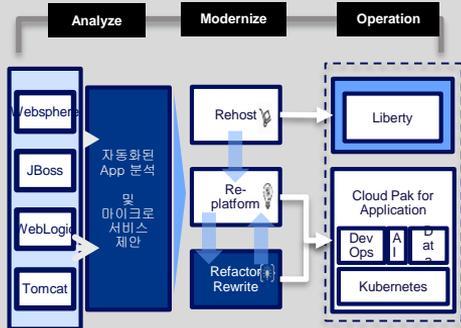
## DevOps

클라우드 환경에 특화된 개발과 운영 환경은 어떻게 구성할지 고민

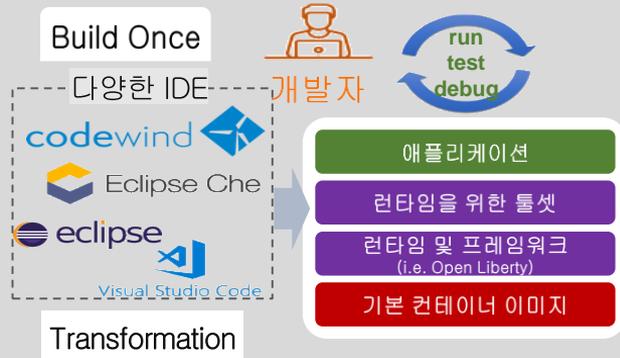
개발된 APP을 운영 환경에 신속하게 빌드/배포하는 환경은 어떻게 구성할지 고민

## Build Once, Deploy Anywhere와 자동화를 통한 신속한 배포

### 기존 앱의 현대화 가이드



### 클라우드에 적합한 개발 환경



### 자동화를 통한 신속한 빌드 / 배포

#### 배포 자동화 파이프라인



IBM Cloud Pak for Apps를 활용한  
**국내 금융권 PoC 수행 사례 - L보험사**

| 2020. Aug

IBM Korea, Cloud Competency Center

고객사에서는 최근 Hybrid-Multi Cloud환경에 있어서, Application Modernization & DevOps의 Core-Technology로 인식되고 있는 'Containerization' 기술 및 방법론을 신속하게 검증하고자 하였음

## Needs

자사 클라우드 도입방향에  
대해서 여러 CSP\* 들의  
제언을 들어보시고,  
내부적으로 검토 및 고민을  
많이 진행해오시면서...

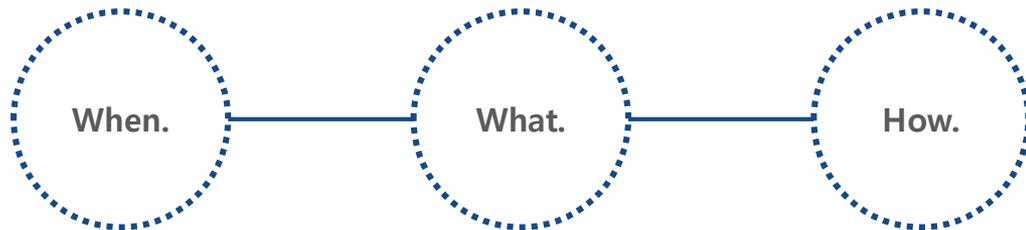
Our Goal is..

- ✓ Vendor **구속이 없는** 클라우드
- ✓ 비즈니스 Agility 를 창출하는 **개발자 생산성 향상** 필요
- ✓ **오픈 소스** 기반 클라우드 플랫폼
- ✓ 단순 cloud component tool 도입으로 끝나지 않는, IT의 **Cloud-Native 내재화** 니즈

\*) Cloud Service Provider

## Feasibility Check를 위한 IBM의 PoC 제안

'빠르게 기술 검증 요소를 Tapping하여 자사 적용 가능성을 확인하고, 클라우드 도입 및 확산을 위한 단계적 추진 방향 및 로드맵을 확보하도록 함'



- 3주간 PoC (Reporting 포함)

- Containerization 실행 및 관리 플랫폼 (OpenShift)
- Cloud로의 Lift & Shift 및 Modernization를 통한 Containerization 방법
- Cloud-Native환경 컨테이너 기반 개발 방법

- 고객사 환경을 고려한, - Red Hat OpenShift 기반 및 WebSphere 연계 App 컨테이너 플랫폼 ( 'IBM Cloud Pak for Apps' )
- IBM Services 리소스 활용 - PoC Lead(1)/ Advisory(1)/ Engineer(1)/ Developer(1)

L생명은 컨테이너화를 추진하는 3가지 전략에서의 제공 가치를 직접 체득함으로써 내재화를 시작하였음

## 1 기존 앱의 Lift & Shift

### 목적

As-Is WAS 환경위에서,  
기존 앱 구조 변경없이  
Lift-Shift로 컨테이너화 간편 수행

### 검증 포인트

'기존 WAS환경을 동일 유지하면서,  
Cloud 환경으로  
바로 Lift & Shift할때,  
얼마나 쉽고, 간편하게 컨테이너화  
및 마이그레이션이 가능한가?'

### Values

- Lift - Shift형태로의 Containerization을 통한, 기존 L생명 보유앱 변환 용이성 및 Complexity 사전 예측
- To-Be Containerization 가용 범위 확정 가능

## 2 기존 앱구조의 현대화 (Modernization)

대표 WAS 환경을 변경하며,  
진행되는 기존 앱의 현대화 과정  
을 통한 컨테이너화 과정을  
개발자 관점에서 확인

'기존 Code를 '현대화'를 전제로  
Refactoring하면서  
컨테이너화 할때,  
Transform과정은 어떻게 되는가.  
개발자에게 주는 Insight는?'

- 앱 현대화과정을 도와주는 소스 분석툴 'Transformation Advisor' 활용성 확인
- 샘플앱의 분석 과정 및 Modernize Insight 추출, Containerization 용이성 확인
- As-Is L생명 앱 현대화 가용 범위 확인

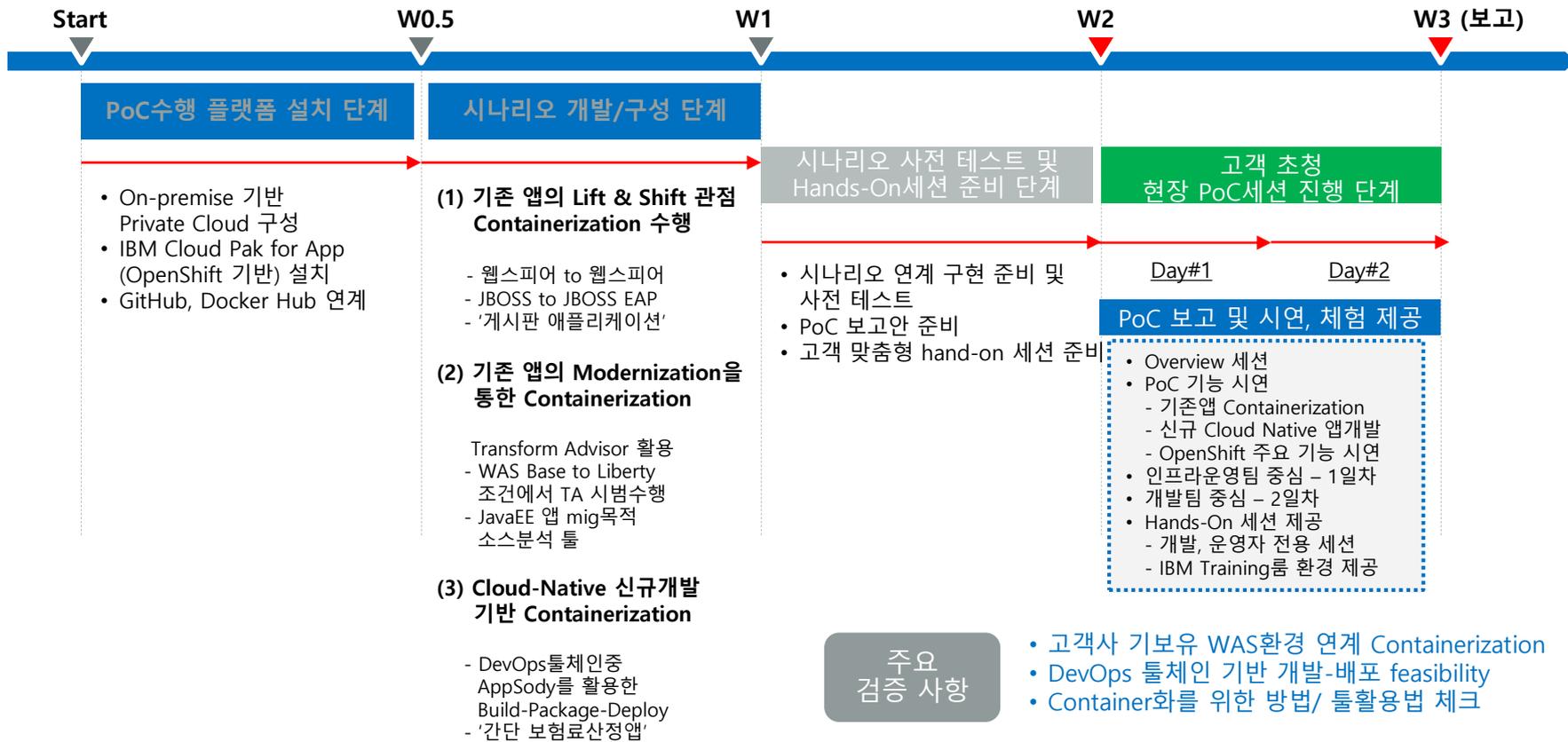
## 3 Cloud-Native로 신규 앱 개발 (Build-Package-Deploy)

Cloud-Native 환경에서, 검증된  
DevOps 개발틀체인으로  
신규 앱의 컨테이너 개발 및  
배포 과정 확인

'DevOps틀체인의 자동화 효과로  
인해  
Cloud-Native 환경에서의  
Refactoring 구현 작업시,  
개발자 용이성, 속도는 어떠한가?'

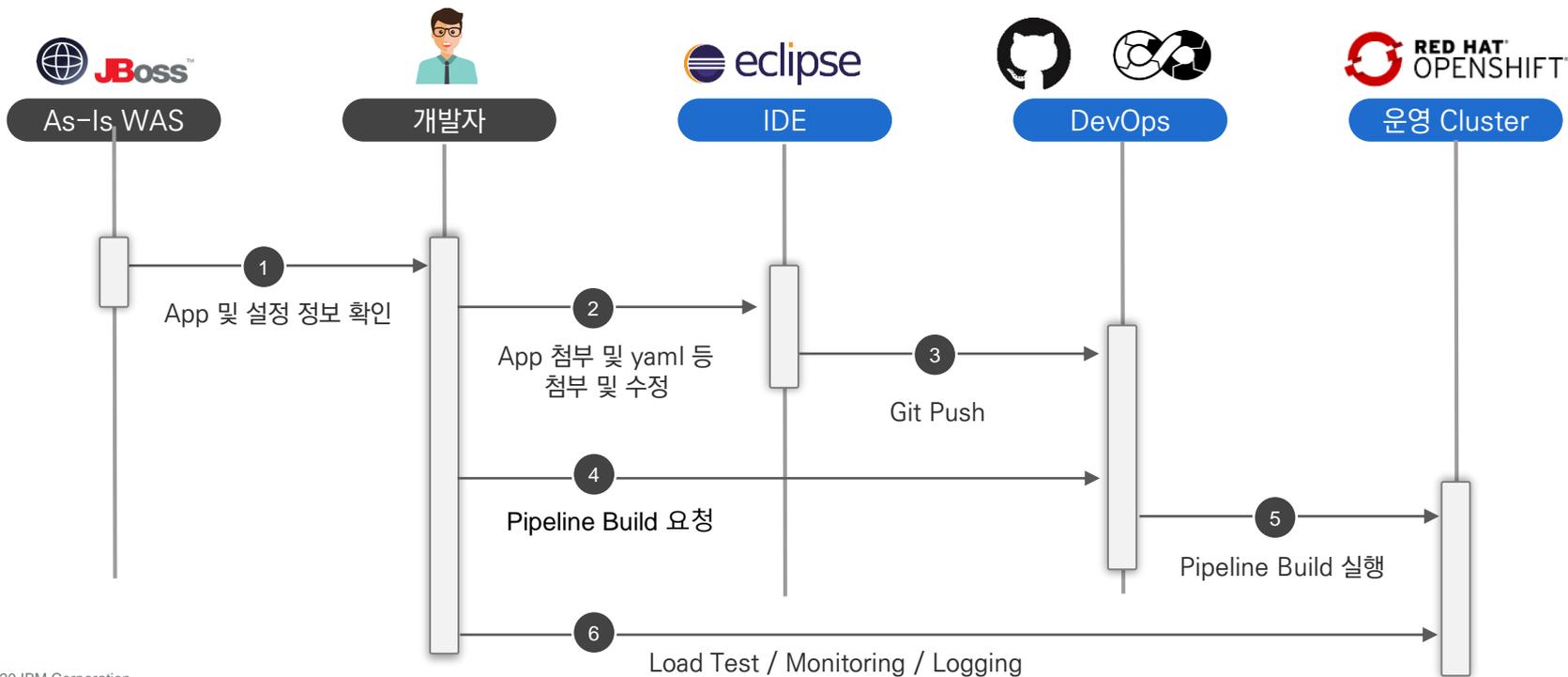
- DevOps 틀체인 도입 및 기존 ITSM연결시, 확보될수 있는 개발생산성 향상 포인트 확인
  - 자동화 적용으로 공수절감 영역 확인
  - 배포시간 단축 범위 예측

IBM은 약 2주간의 준비-구축 작업, 1주간의 PoC 산출물 보고 준비 세션을 가지고 총 3주간 PoC를 제공함

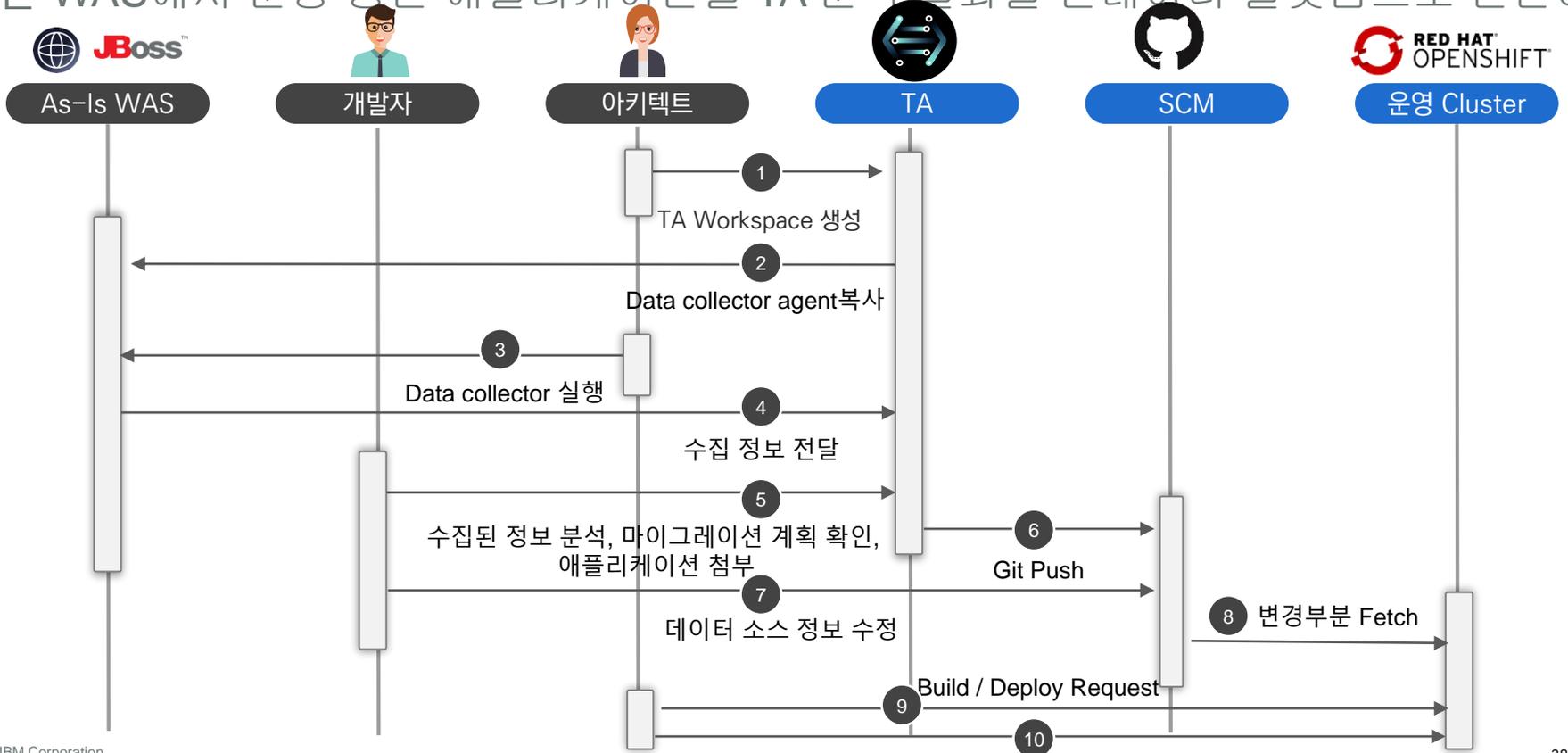


# 기존 WAS에서 운영 중인 애플리케이션을 컨테이너 플랫폼으로 전환하는 절차

## 시나리오 #1 – Websphere / JBOSS Migration



# 기존 WAS에서 운영 중인 애플리케이션을 TA 분석 결과를 컨테이너 플랫폼으로 전환하는



오픈소스 Kabanero의 기업용 버전인 A4T는 CodeReady, Codewind, Appsoy, Knative, is

## A4T 연계 스택

CODEWIND - PLUGIN FOR ECLIPSE, VSCODE			ECLIPSE CHE		
					
MICROPROFILE	NODE JS	OPEN LIBERTY	LOOPBACK	SPRING BOOT	JAKARTA EE
					
APPSODY	KNATIVE	TEKTON	OPERATOR	ISTIO	
					

### PROGRAMMING TOOLS

### PROGRAMMING FRAMEWORKS

JAVA AND NODE.JS TECHNOLOGIES

### CLOUD DEVOPS TECHNOLOGIES

APPSODY CONSISTENT DEVELOPMENT TEMPLATES

KNATIVE SERVERLESS ARCHITECTURES

TEKTON CI/CD PIPELINES

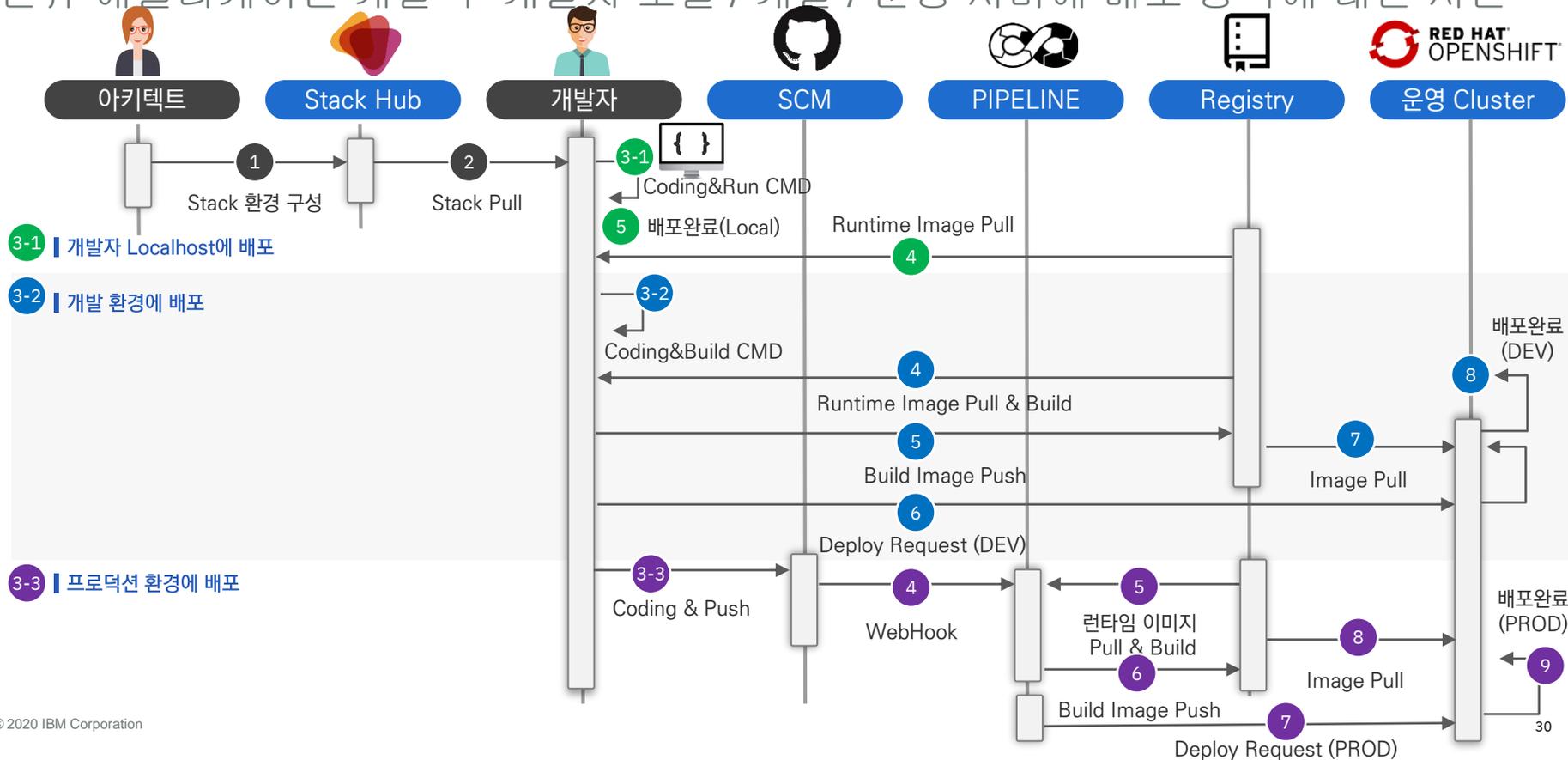
OPERATOR APPLICATION MANAGEMENT

ISTIO MICROSERVICE COMMUNICATION

### CONTAINER PLATFORM

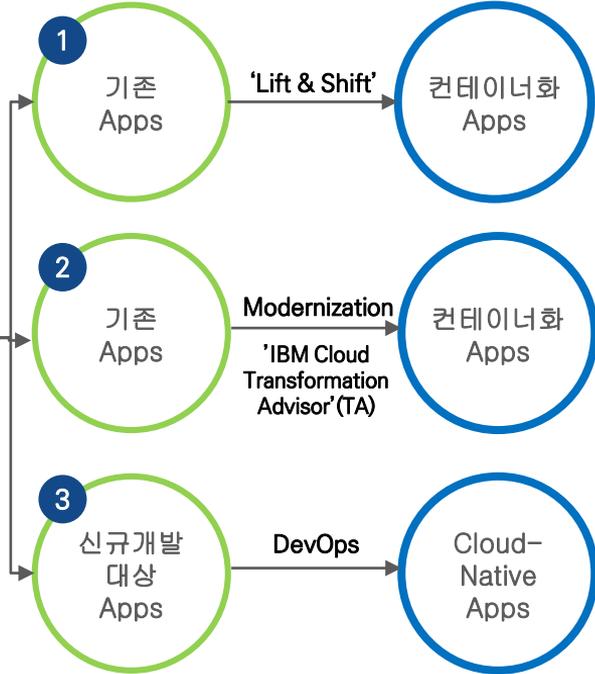
CAN DEPLOY ON KUBERNETES, OR PREFERRED ON RED HAT OPENSIFT

# 신규 애플리케이션 개발 후 개발자 로컬 / 개발 / 운영 서버에 배포 방식에 대한 시연



IBM Cloud Pak  
for App 기반  
컨테이너  
오케스트레이션  
플랫폼

## Containerization 3 Approaches



## 수행 Actions

- Replatform 방식  
컨테이너라이제이션
- Git-pipeline추가-Image 배포
- 기존 WAS환경에서 컨테이너화  
- WAS to WAS  
- JBOSS to JBOSS EAP
- Apps Modernization
- 컨테이너화를 위한, 소스코드 분석 및 자동화 도구 활용('TA')
- Code 수정 부분 인사이트 추출
- Cloud-Native환경기반 DevOps 틀체인 활용  
앱개발-패키지-Deploy (Appsofy Stack활용)
- 오픈소스기반 CI/CD Pipeline 'Tekton'활용 자동 배포 수행

## 검증 포인트 질문

→ 마이그레이션 용이성 및 적용범위

→ 분석/변환 가시성 개발자 활용성

→ Cloud-Native 구현 용이성 및 속도 CI/CD 자동화 체감

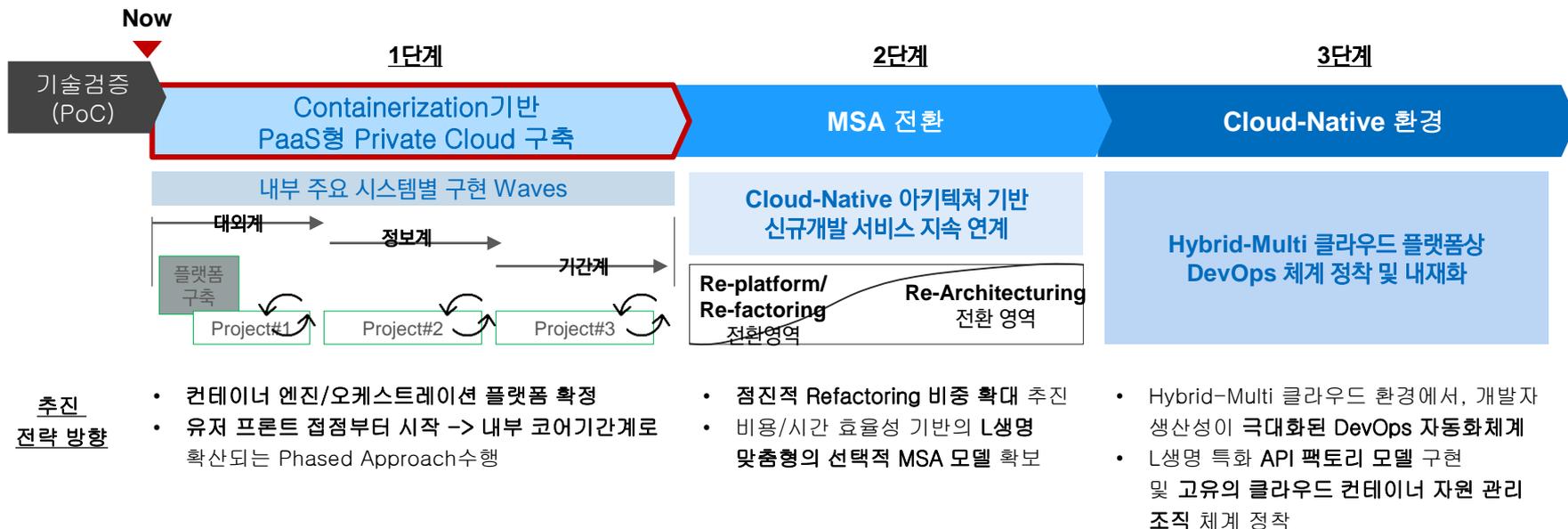
## 시사점

- 오픈소스 기반 WAS 마이그레이션 동반 컨테이너화 용이
- **컨테이너화 적합성 체크후 우선 수행요**

- L생명 내부, JavaEE기반 서비스, Springframework 기반 애플리케이션 선적용 가능
- **TA활용을 통한 전환 개발 공수 절감**
- **컨테이너화 가속화 지원**

- 향후 신규개발 공수 절감 및 리드타임 감소
- **DevOps 개발/운영환경 기본 프로세스 확보**
- **MSA로 가기위한 Short-Cut 환경 구축**

PoC를 통해 확인된 ‘컨테이너라이제이션’ 이라는 차세대 클라우드 코어 기술을 시작으로, 단계적 추진 Approach를 통해 자사 고유의 MSA 및 Cloud-Native 플랫폼 환경을 안정적이고 혁신적으로 확보함



오픈시프트 컨테이너 오케스트레이션 기반 DevOps 튜체인 환경

IBM®