

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

hongseok@kr.ibm.com
IBM Automation, Integration



클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

2019년 까지 전세계 기업의 90% 는 “클라우드“ 를 이용했지만¹, 워크로드의 20% 만이 클라우드 환경으로 이동하였습니다.²



Application 현대화는 어렵습니다.

-기업은 레거시 애플리케이션을 클라우드 네이티브로 다시 작성하려고 했지만, 비용이 많이 들고 비즈니스 가치를 빠르게 창출하지 못했다는 사실을 알게 되었습니다.



모두를 만족시키는 단하나의 방법은 있을 수 없습니다.

- 현재 기술/애플리케이션 환경과 혁신의 비즈니스 목표를 기반으로 다양한 현대화 처리가 필요합니다.



Application 현대화의 다음단계는 통합입니다.

- Business Logic (Application)에 대한 초기 초점에서 이제 Data 및 Integration을 포함하는 보다 전체적인 초점으로 이동하는 것입니다.

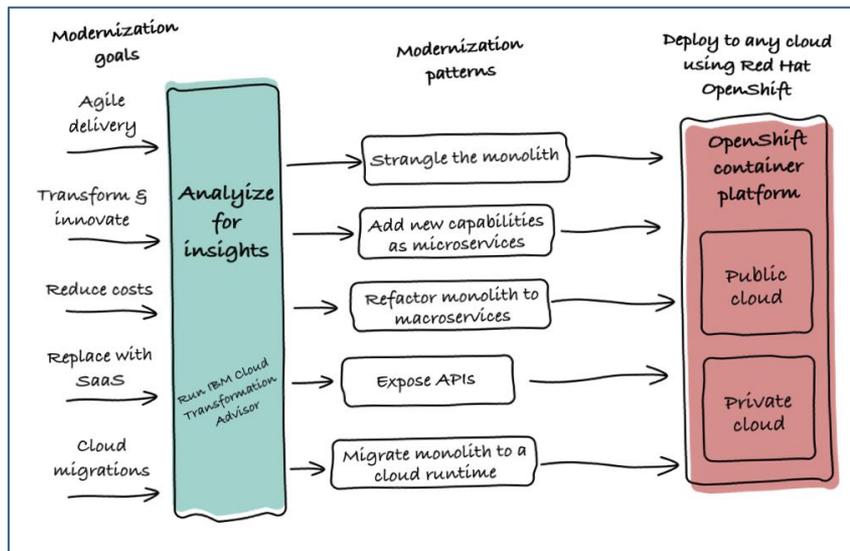
Source: “[1]: Brooks, Carl. “What’s on the Minds of Cloud-Focused CTOs in 2018?”. Accessed April 2020. <https://go.451research.com/what-is-on-mind-of-cloud-focused-CTOs.html>

[2]: Keverian, Ken, Arvind Krishna, Steve Robinson, Anthony Marshall. “Next-generation hybrid cloud powers next-generation business.” IBM Institute for Business Value. August 2019. <https://www.ibm.com/thought-leadership/institute-business-value/report/hybrid-cloud>

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

What is modernization ?

- 기존 애플리케이션이나 워크로드를 컨테이너화 하는것 ?
- 애플리케이션을 마이크로서비스로 리팩토링 ?
- 모노리식 앱을 점진적으로 마이크로 서비스로 교체 ?
- 완전히 새로운 마이크로 서비스로 재개발 ?
- 배포 프로세스의 자동화 ?
- 클라우드 환경으로 Lift & Shift ?
- 앱의 기능들을 API 로 노출 시키는 것 ?
- 새로운 마이크로 서비스들로 낡은 코드들의 기능을 강화 하는것 ? API , Service (AI , Data Science)
- 우리 데이터는 ? !!!!!!!
- 클라우드와 클라우드간 애플리케이션의 통합은 ?



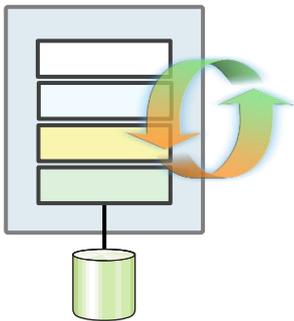
<https://www.ibm.com/cloud/garage/content/field-guide/app-modernization-field-guide/>

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

What is Microservices ?

Monolithic

모든 기능이 하나의 "응용프로그램"으로 구현



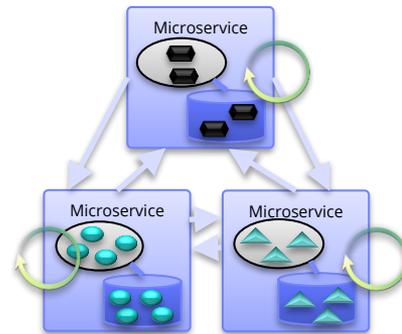
- Tightly Coupled
- 적은 수의 대형 서버
- 장기 프로젝트
- 몇년에 한번 업데이트

Greenfield (신규 앱)



Microservices

각 기능은 별도의 "서비스" 로 구현



- Loosely Coupled
- 많은 수의 소형 서버
- 베타에서 공개
- 단기, 빈번한 업데이트

1. Scale
2. Resiliency
3. Distributed
4. Agility
5. Flexibility

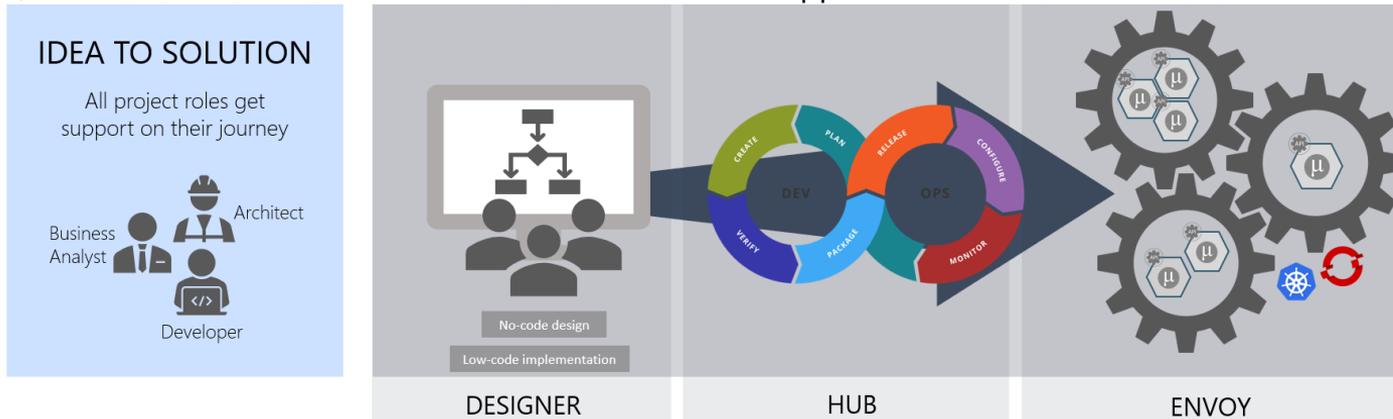


Brownfield (기존 앱)

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

IBM ISW - Overview

IBM Solution Workbench(ISW)는 비즈니스 분석가(설계자)와 개발자가 Domain Driven Design(DDD)의 원칙에 따라 Microservice를 공동으로 설계하고, 빌드, 패키징, 테스트 및 배포하는 기능과 Cloud-native 환경에 필요한 보안, 데이터와 이벤트 처리, 인프라의 관리 등의 Pre-Built된 서비스를 제공하는 Application 현대화 도구입니다.



Solution Designer를 사용하면 비즈니스 분석가, 개발자 및 설계자가 Domain Driven Design(DDD)의 원칙에 따라 Microservice를 공동으로 설계할 수 있습니다.

Solution Hub는 기존 개발 프로젝트 및 환경을 추적하는 중앙 집중식 레지스트리입니다. 또한 미리 정의되었지만 확장 가능한 파이프라인을 기반으로 Microservices Designer에서 생성된 솔루션을 빌드, 패키징, 테스트 및 배포하는 기능을 제공합니다.(CI/CD)

Solution Envoy는 최신 마이크로서비스 환경에 필요한 모든 서비스(모든 kube 리소스, 보안, 데이터와 이벤트 인프라의 관리)를 제공합니다. 솔루션의 대상 런타임 내에 설치되며, 진일보한 모니터링 기능을 지원하고, 해당 런타임 내에 배포된 솔루션에 대한 공유 서비스를 제공합니다.



클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

Low-code : Domain-Driven Design을 활용하여 비즈니스 요구 사항을 최신 기술의 마이크로서비스 구조로 자동 생성

- 마이크로 서비스는 비즈니스 도메인 중심의 설계 사상인 Domain Driven Design(DDD)과 밀접한 관련이 있습니다. ISW는 DDD를 기반으로 하는 마이크로 서비스 개발을 지원하기 위해 전용 **low-code** 프로젝트 생성 기능을 제공합니다.
- 공동 작업을 위한 디자인 환경(Solution Designer)에서 도메인 모델링을 지원
 - **entities, aggregates, commands, services, events, event agents** 등의 마이크로서비스 빌딩블록 코드를 자동 생성
 - API 및 이벤트 인터페이스를 통해 다른 업무 도메인 또는 기술 하위 도메인을 격리시켜 Decouple을 보장

The screenshot displays the IBM Solution Designer interface for configuring a 'Root Entity InvestmentProject'. The main window is divided into several sections:

- Navigation:** Overview, APIs, Domains, Integrations, CI/CD.
- Entity Level:** Documentation for 'core:InvestmentProject'.
- Associated Properties:** A table listing properties for the entity level.
- Associated Commands:** A list of commands for the entity level.
- Diagram:** A domain model diagram showing relationships between entities like 'InvestmentProject', 'InvestmentAmount', 'FinancingObjects', etc.

Name	Label	Property Definition
investmentAmount	Amount	core:investmentAmount
financingObjects	Financing objects	core:financingObjects
name	Name	core:name
projectType	Project type	core:projectType
projectStatus	Project status	core:projectStatus
owner	Owner	core:owner
description	Description	core:description
financingProposals	Financing proposals	core:financingProposals
planCovenants	Plan covenants	core:planCovenants
planCollaterals	Plan collaterals	core:planCollaterals
calculation	Pricing	core:calculation

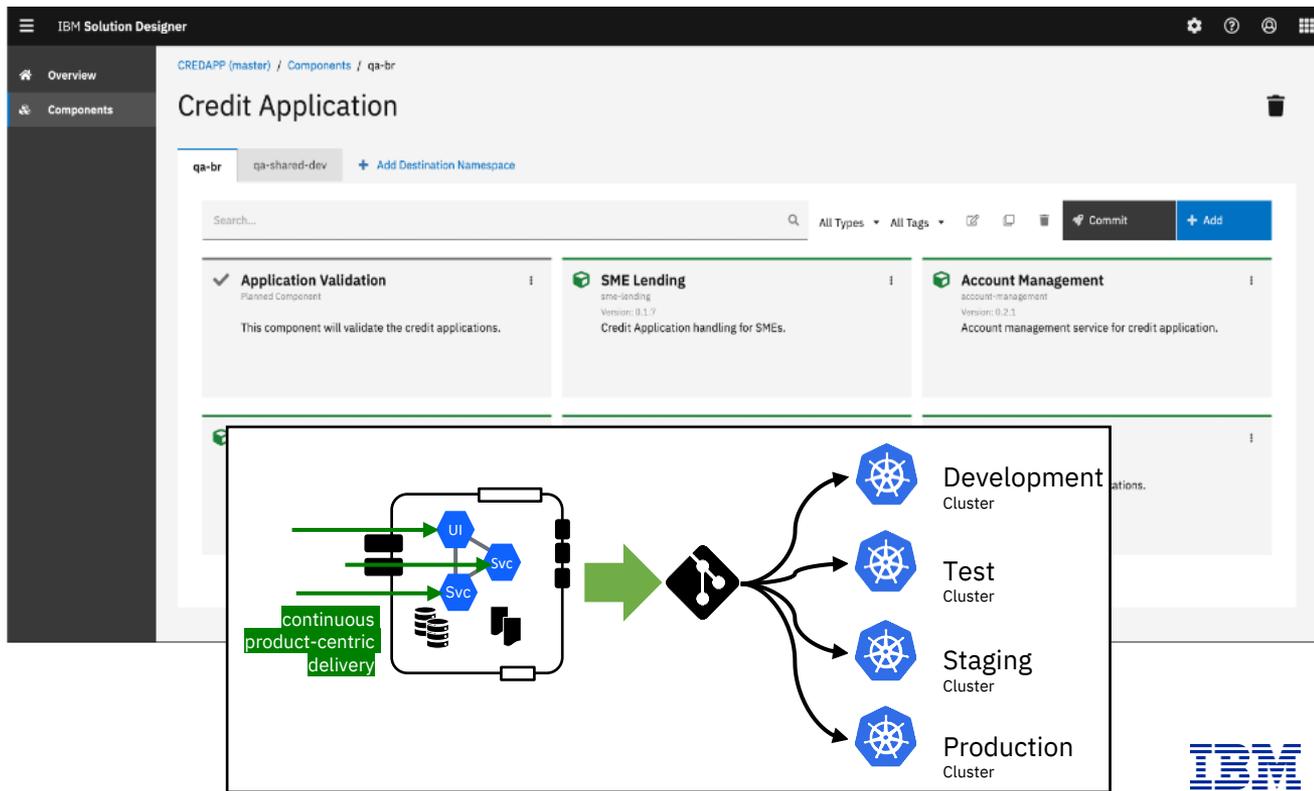
Associated Commands at this Level:

- Add collateral
- Add covenant
- Add product
- Roll proposal
- Create project
- Initiate Project
- Initiate proposal
- Retrieve collateral
- Remove covenant
- Remove product
- Update collateral
- Update covenant
- Update product
- Update project
- Update proposal

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

다양한 비즈니스 Application들을 이용해서 쉽게 조합하는 Composable Solution 제공

- Application은 독립적인 비즈니스 기능의 구성 요소입니다.
- ISW는 애자일 팀이 GitOps의 지속적인 제공을 기반으로 조합 가능한 비즈니스 솔루션을 제품 형태로 지속적인 제공을 지원합니다.
- 모든 솔루션 정의, 구성 및 환경은 선언적이며 버전을 제어할 수 있습니다. 응용 프로그램 배포 및 수명 주기 관리는 자동화되고, 감사가 가능하며, 투명하게 관리됩니다.
- 비즈니스 솔루션은 도메인 또는 요구 사항에 따라, 하나 이상의 응용 프로그램으로 분해될 수 있습니다.



클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

IBM Mono2Micro - Overview

Opportunity

고객사에서 관리하고 있는 여러 애플리케이션중, Migration 을 통해 클라우드에 적합한 형태로 전환하여 컨테이너/클라우드화에 따른 개발 / 운영상 이점을 확보

문제점

- 기존 코드의 분석에 높은 이해와 분석 능력을 필요
- 반복적인 작업으로 인한 시간 / 비용 과다 소요
- 아키텍처의 재구성이나 재구현을 필요로 함

요구사항

- ✓ 프로젝트를 진행할때 수작업의 범위를 최소화
- ✓ 자동화된 방안으로 마이크로서비스 를 가이드
- ✓ 프로젝트 진행에 필요한 각종 서비스 코드를 자동으로 생성
- ✓ 마이그레이션 프로젝트의 비용을 절감 하는 방안

앱 현대화를 위한 컨테이너화 제안

Mono2Micro

AI based automatic transformation of monoliths into microservices



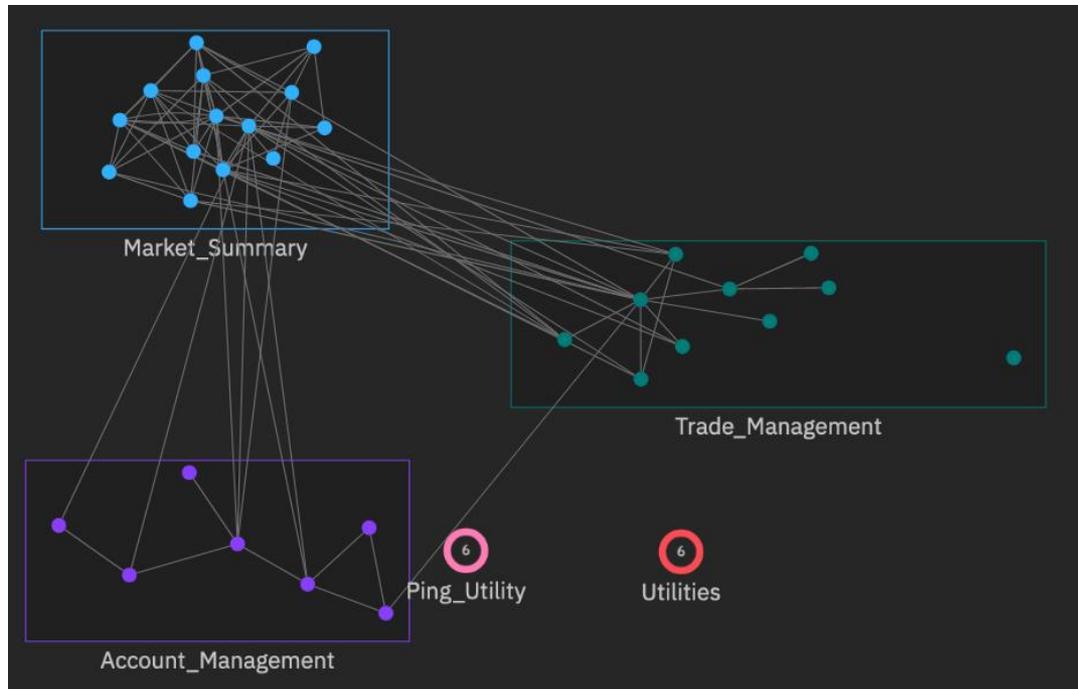
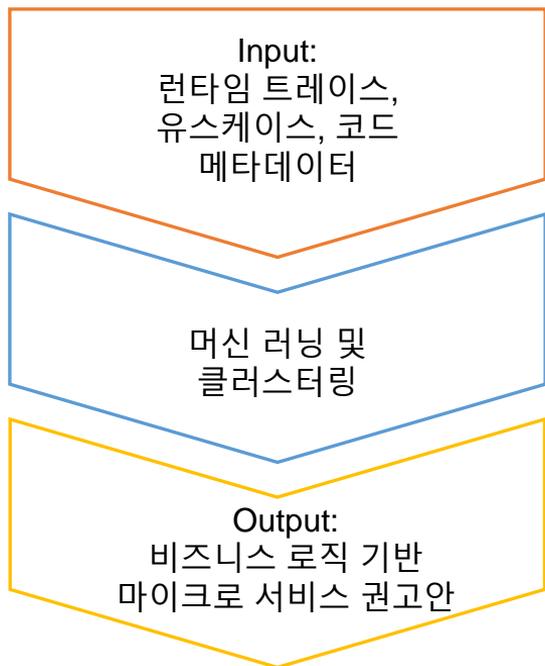
아키텍트 가 리팩토링을 수행하기 위한 다양한 기능을 통해 수작업에 따른 각종 부작용과 시간, 비용을 절감

자동화된 마이크로 서비스 권고 사항 이행

데드코드를 식별하여 마이크로 서비스 이행을 가속화

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

마이크로 서비스 후보군 추출



클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

자동 코드 생성 기능

Input

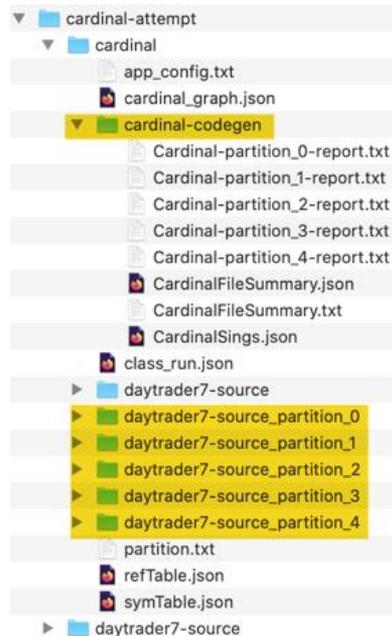


마이크로 서비스 권고사항
(비즈니스 로직) +

원본 소스 코드

각 마이크로 서비스에 대한 코드베이스

Output

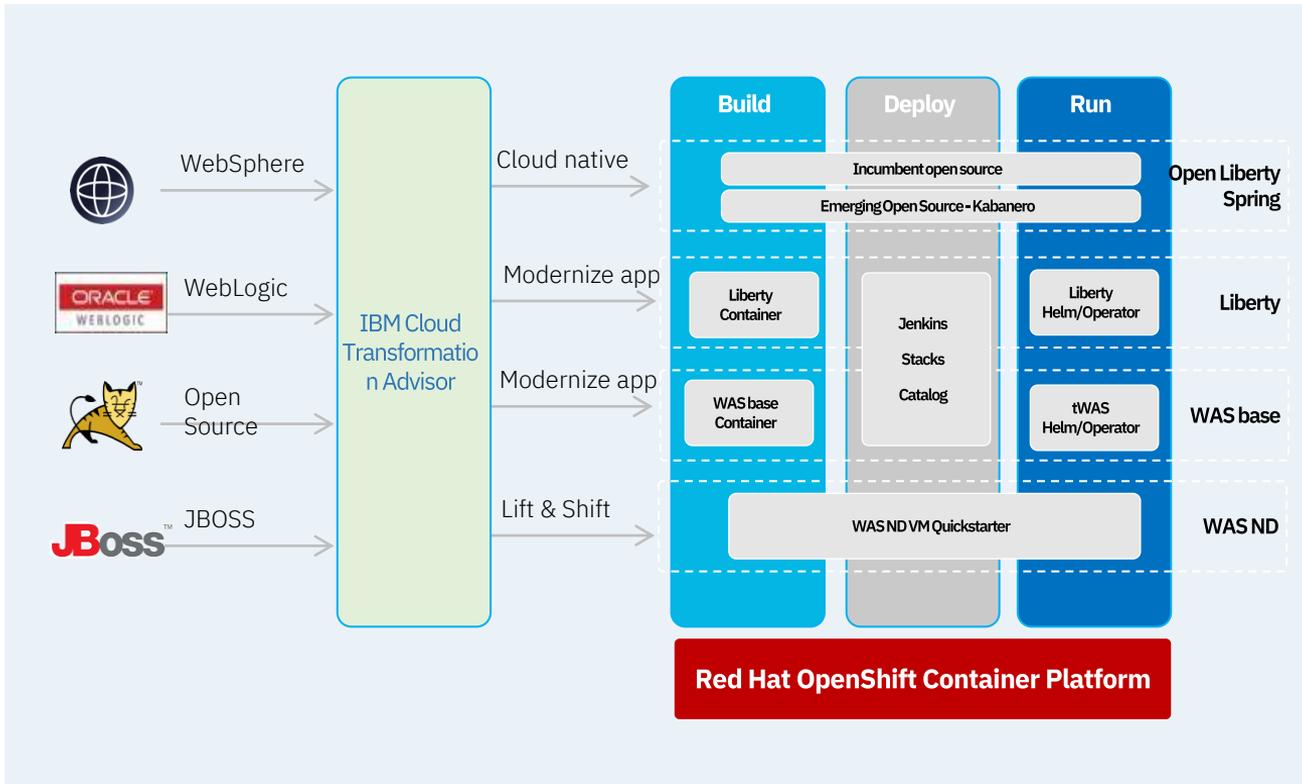


자동 생성된 마이크로 서비스 코드 예시

```
public Collection<QuoteDataBean> getTopGainers() {  
  
    Form form = new Form();  
    form.param("referenceId",  
String.valueOf(this.referenceId));  
  
    JsonObject response_json =  
client.target(DAYTRADER_BIZLOGIC_REST_URL)  
.path("getTopGainers")  
.request(MediaType.TEXT_PLAIN)  
.post(Entity.entity(form,  
MediaType.APPLICATION_FORM_URLENCODED_TYPE), String.class);  
String response =  
(String)response_json.get("return_value");  
  
    Collection<String> response_decoded =  
(Collection<String>)SerializationUtil.decode(response);  
String ret_type = (String)response.get("return_type");  
    . . .  
}
```

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

IBM Transformation Advisor - Overview



다양한 런타임 지원

WebSphere Application Server

- WebSphere ND
- WebSphere Base
- Liberty Core
- JBOSS Enterprise Application Server**

설명

- 클라우드 네이티브 및 기존 환경에 최적화된 프레임워크 및 런타임 세트를 제공
- 오픈 소스, 공통 서비스, 원하는 개발자 툴, 통합 DevOps를 시작으로 컨테이너가 포함된 클라우드 네이티브 앱 개발



IBM WebSphere Liberty Overview

Supports Java EE, great for monoliths and microservices, ideal for modernization

Why Liberty?

보다 높은 확장성
보안성 개선
낮은 오버 헤드

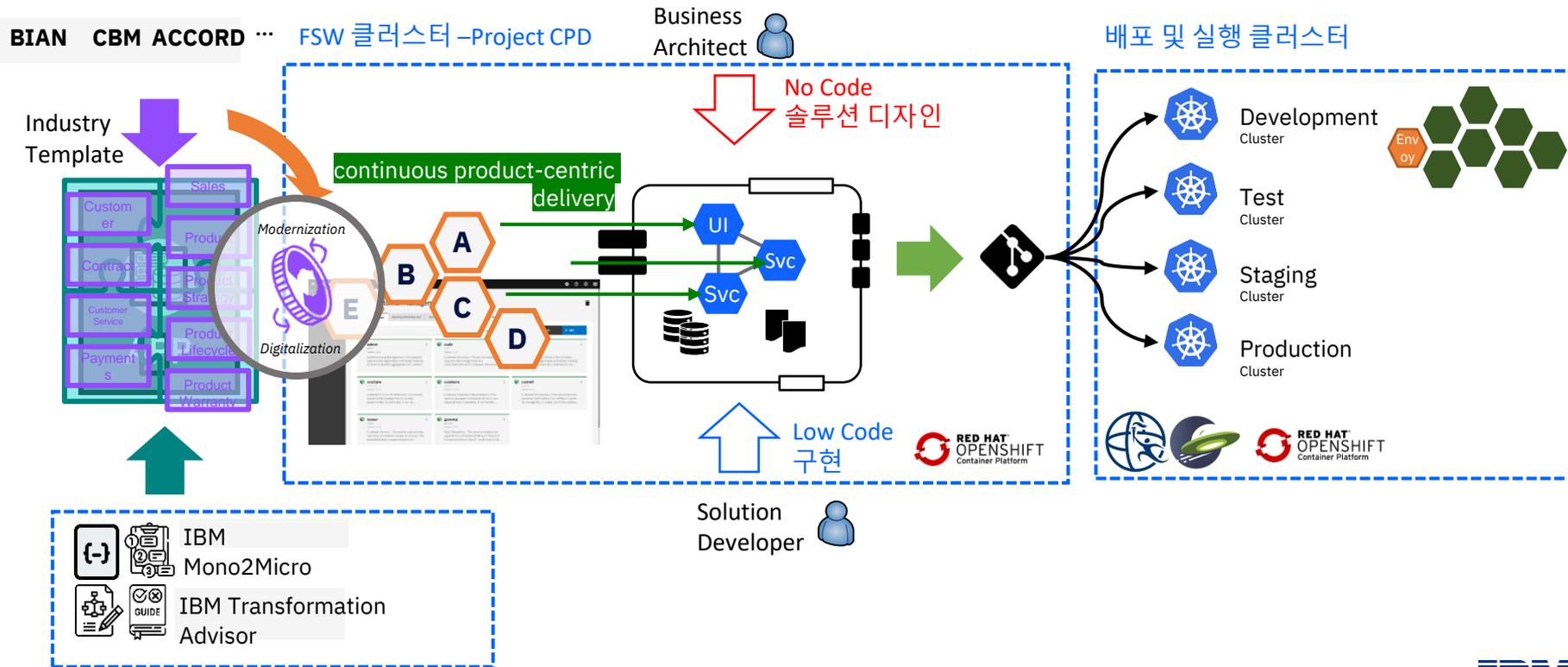
3년간 158% ROI



*The Total Economic Impact™ Of IBM WebSphere Liberty: A Forrester Total Economic Impact™ Study Commissioned By IBM September 2018 - ibm.com/downloads/cas/NVY3KY4E

클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

Application Modernization & Transformation



클라우드와 마이크로 서비스를 위한 애플리케이션 현대화

디지털 전환을 통한 비즈니스 성과 개선 효과



9.5 X

제품 품질 (product quality)의 개선 가능성 향상



5 X

비용절감 (cost reduction)의 개선 가능성 향상



1.7 X

앱 이용자 및 직원 만족도 (user/employee satisfaction) 개선 가능성 향상



1.6 X

고객 유지 (customer retention) 개선 가능성 향상

현대화는 비즈니스 성과를 개선 시키는 가속기

IBM