

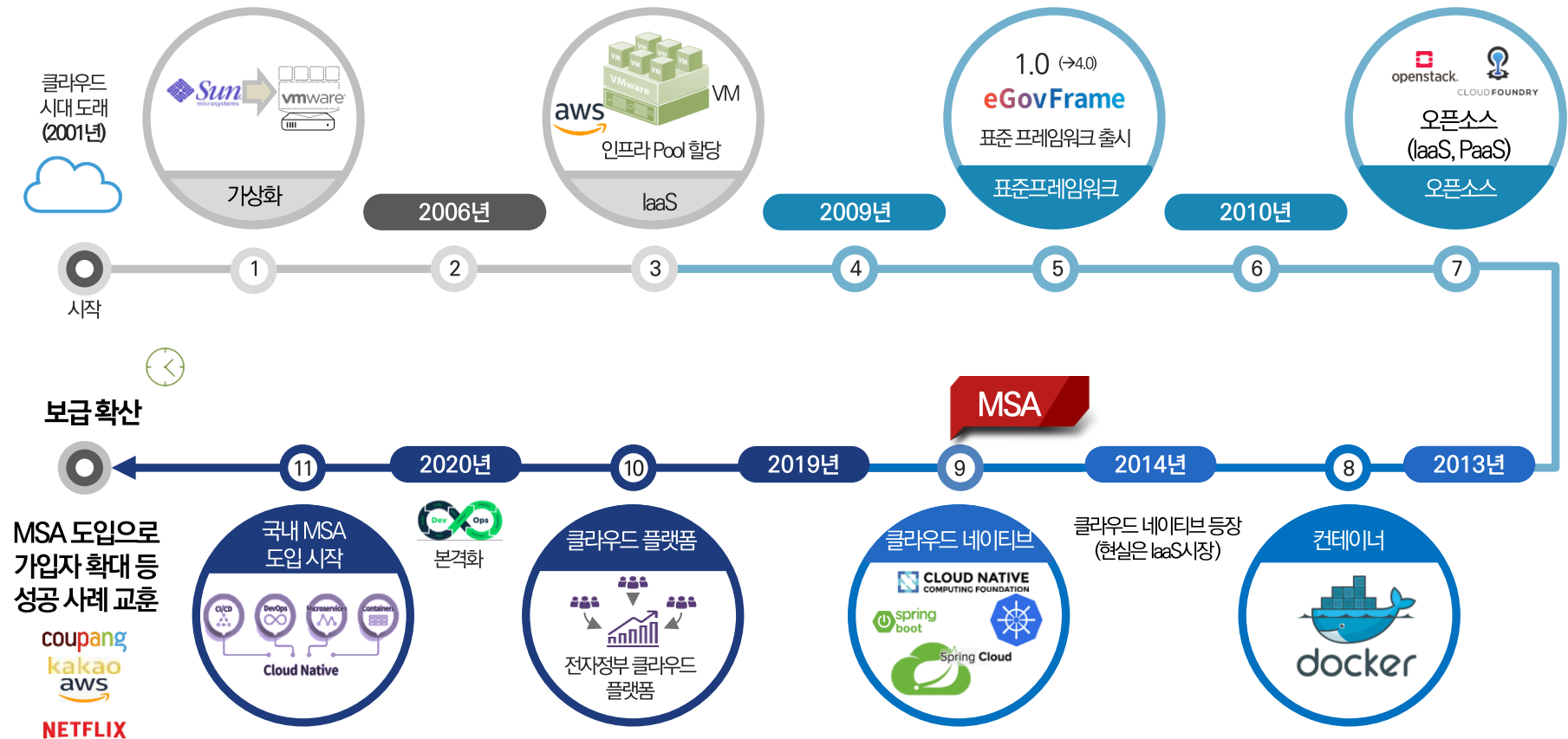
클라우드 네이티브 기반 행정·공공 서비스 확산 지원

클라우드 네이티브 온라인 설명회

발주자 가이드 (발주 안내서)



클라우드 네이티브 발자취



*클라우드 네이티브 : MSA(Micro Service Architecture 약어)로 불리우며, 마이크로서비스 아키텍처 스타일은 단일 응용 프로그램을 자체 서비스로 실행하고, 경량 메커니즘(HTTP)으로 통신하는 작은 서비스 모음으로 개발하는 접근 방식임 (Martin Fowler)

발표 목차

I 클라우드와 클라우드 네이티브는 왜 필요한가?

II 클라우드 네이티브는 어떤 특성을 가지고 있는가?

III 클라우드 네이티브는 어떻게 발전하고 있는가?

IV 공공 클라우드와 클라우드 네이티브는 정부정책을 추진합니다.

V 어떤 공공업무에 클라우드 네이티브의 적용이 가능할까요?

VI 클라우드 네이티브 도입 고려사항은 어떻게 되나요?

VII 클라우드 네이티브 도입 절차는 어떻게 되나요?

VIII 클라우드 네이티브 도입 비용산정은 어떻게 해야 하나요?

IX 클라우드 네이티브의 아키텍처 참조모델은?

X 클라우드 네이티브는 구성요소는?

XI 공공 클라우드 네이티브는 어떤 효과를 제공하게 되나요?

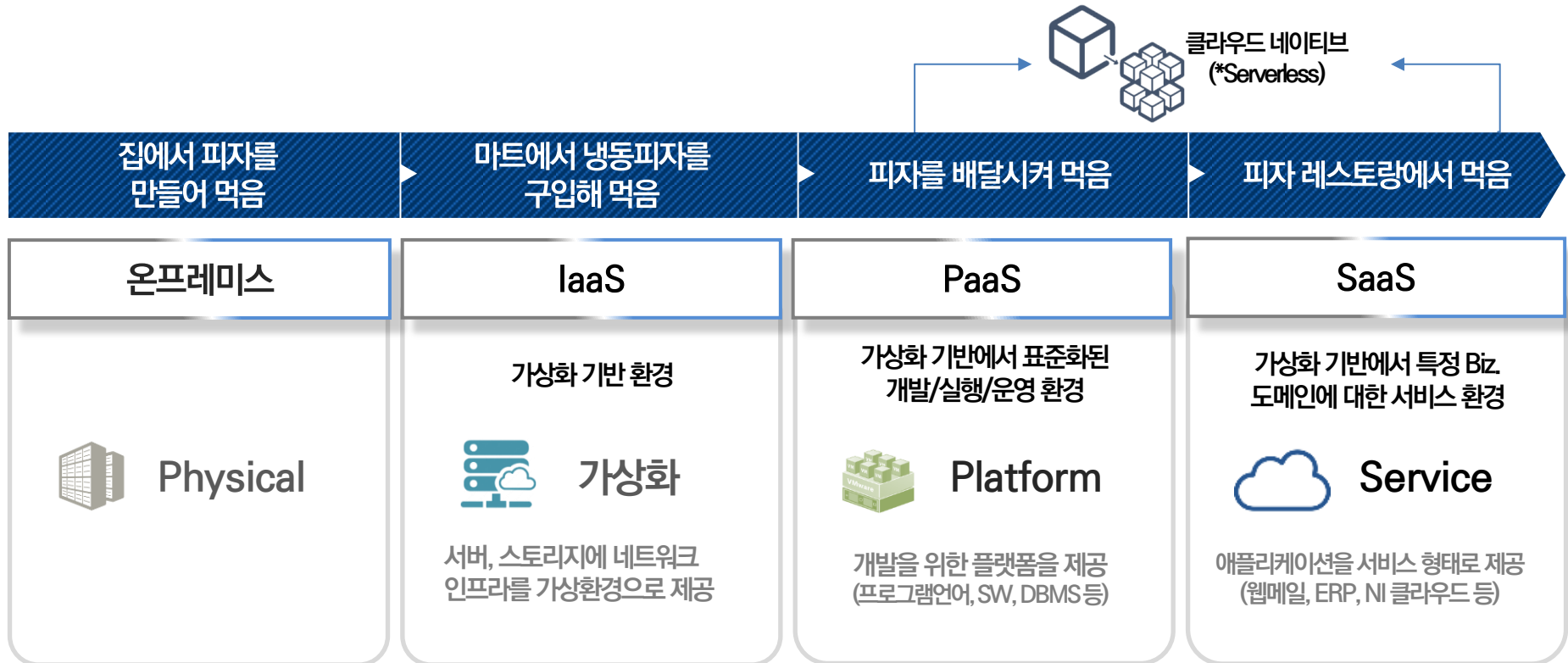
클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드와 클라우드 네이티브는 왜 필요한가?



클라우드는 무엇인가?

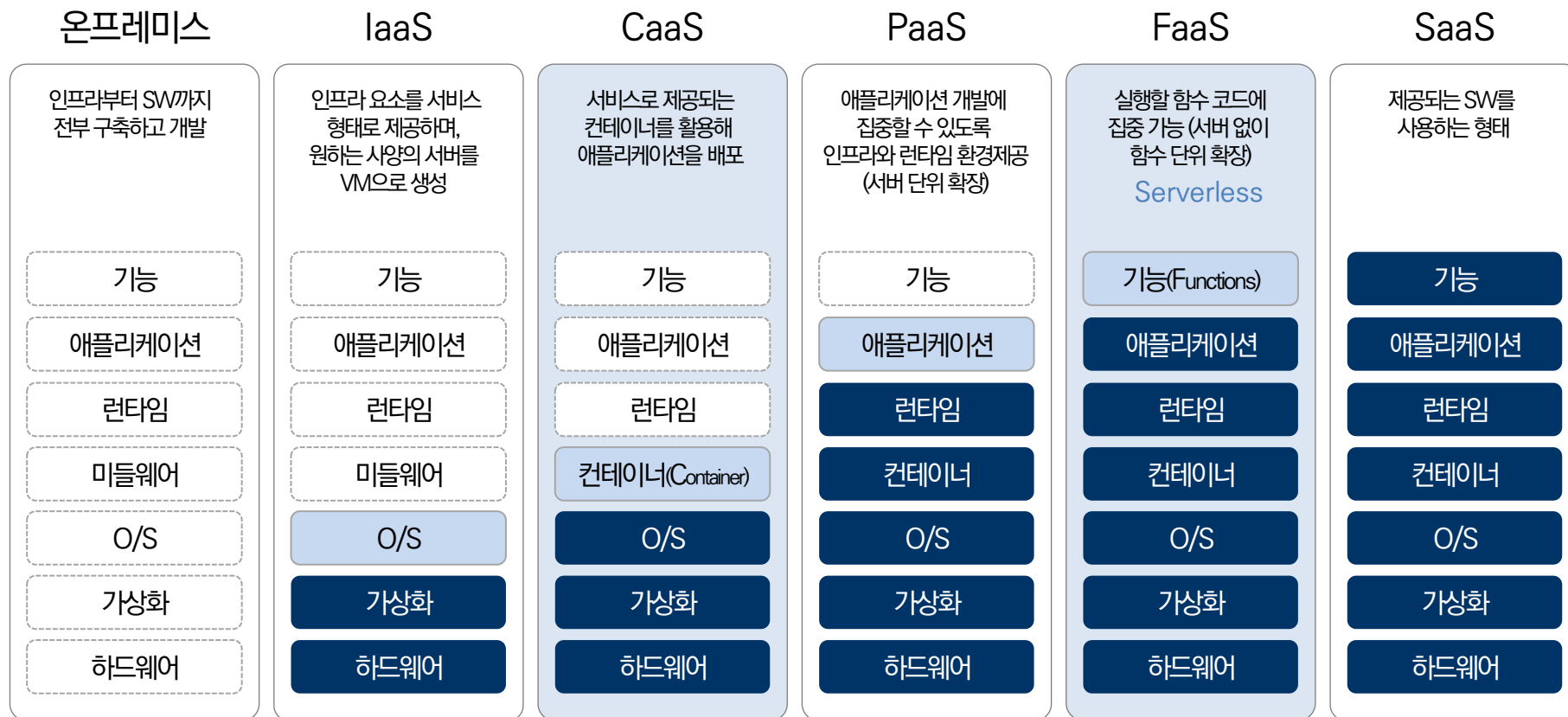
가상화된 컴퓨팅 자원을 활용하여 소프트웨어, 플랫폼, 인프라 등의 IT 서비스를 제공하는 기술이며, 기존 IT 대비 효율적이고 신속한 서비스 제공이 가능함



* **Severless** : 서버리스 컴퓨팅은 서버가 존재하지만, 사용자의 운영 개입 없이 클라우드 플랫폼에서 자동으로 관리되는 서비스로, 애플리케이션 함수 실행 서비스 및 백엔드 서비스를 포괄

클라우드 서비스 모델

클라우드 서비스는 IaaS, PaaS, SaaS가 일반적으로 제공되는 형태이며, 컨테이너와 서버리스 등의 최신 기술이 발달되면서 CaaS와 FaaS 등 다양한 클라우드 서비스 모델로 세분화되고 있음



*범례:



이용기관 직접 준비




이용기관, 공급자 공동책임



공급자 제공

클라우드 네이티브 성공사례

Netflix는 클라우드 네이티브 성공사례로, 전세계 확산 서비스를 제공하여 사용자에게 고품질 동영상 서비스를 안정적으로 제공



넷플릭스는 매년 전세계 인터넷 트래픽의
15% 이상을 차지하고 있음

2021년에는 이미 2억 400만명
(3년 만에 2배 증가)의 구독자가 생겼고,
매 4분기마다 5백만명이
200개 이상의 국가에서 구독 중

NETFLIX
OSS

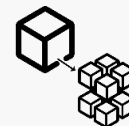
Netflix
OpenSourceSoftware
Center



기술팀이 8년간 노력해온 결과



인프라를 자체 센터에서
Public 클라우드 이전



모놀리틱 프로그램을 작게
관리할 수 있는 마이크로서비스
아키텍처로 변경



애플리케이션 함수 실행
서비스하는 서버리스 컴퓨팅 및
백엔드 아키텍처를 제공

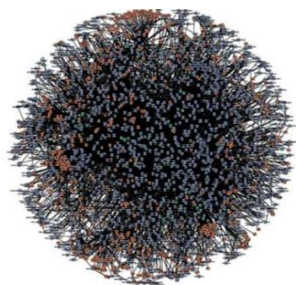
클라우드 네이티브 국내·외 선도 도입 사례

AWS

빠른 배포 구현

수 천개 팀(자율적 DevOps팀) X 마이크로서비스 아키텍처
X 지속적 배포(CD) X 다양한 개발 환경

수 천개 팀

지속적
배포(CD)마이크로서비스
아키텍처다양한 개발
환경

넷플릭스

가입자 대상서비스 확대

Netflix Open Source Software Center

클라우드
실행

개발조직

You Build it

개발·운영



DevOps 직원

You Run it

운영환경
배포

개발자

You Support it

선진
사례

KAKAO

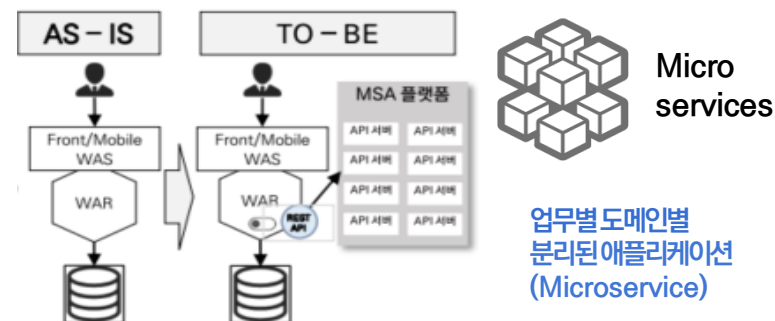
계열사 신규서비스 확대 및 빠른 출시 사례

카카오의 애자일 문화, 일하는 방식 관리를 위한
전담팀 및 개발플랫폼 운영



11번가

서비스 분리를 통한 점진적 MSA 전환

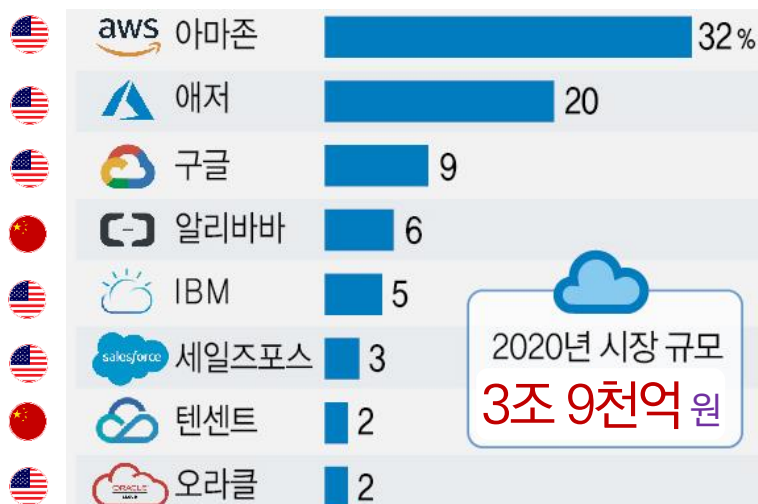


국내·외 클라우드 시장 (1/3)

현재 국내·외 클라우드 시장은 외산이 주도
다양한 서비스, 손쉬운 개발·배포 환경으로 외산 클라우드 선호, 국내 퍼블릭 PaaS 성장中

국내 클라우드 시장의 외산 장악

“국내 클라우드 서비스 시장 점유율”
 - 2020년 4분기 -



kt NAVER 국내기업 점유율은 미비

출처 연합뉴스 <https://m.yna.co.kr/view/GYH20210205000400044>

국내 퍼블릭 PaaS 시장 급격한 성장전망

“2020년 클라우드 부문 매출 현황”

NAVER	2,737억 원	40% 상승 ↑
kt	4,000억 원	20% 상승 ↑
이-이-이	1,600억 원	250% 상승 ↑

가비아, 스마일서브 등이 열심히 사업 진행 중

일반 기업 관련 동향



6,000억 중
AWS : 자사 클라우드 = 6:4

국내·외 클라우드 시장 (2/3)

현재 국내·외 클라우드 시장은 IaaS에서 PaaS로 중심축 이동 중

개발자를 위한 플랫폼 서비스, PaaS

경쟁력 있는 PaaS가
SaaS 및 IaaS 점유율 확대의 원천

운영 체제

실행 환경

개발 환경

분석 환경

데이터 베이스

웹 서버

클라우드 서비스의 차별성 PaaS가 결정

글로벌 PaaS 시장 규모 2026년까지
연평균 19.6%로 성장 예상

2020년
약 67조 원

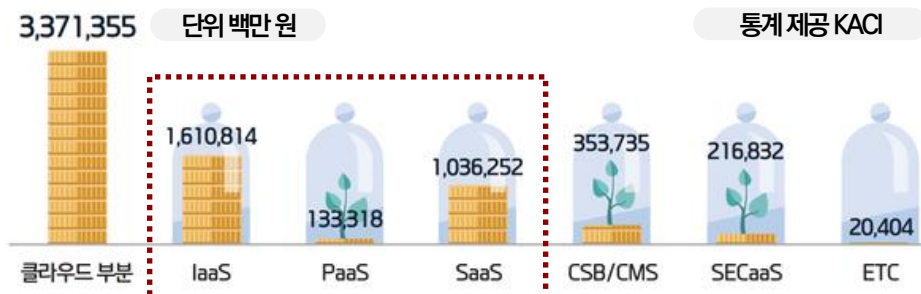


2026년
약 196조 원

출처: '2020년까지 서비스로서의 플랫폼 시장' 보고서 MarketsandMarkets

국내 PaaS 시장의 무한한 가능성에 주목해야 할 시점

2019년 국내 클라우드 서비스 부문별 매출 현황



IaaS와 SaaS
성장세는
이미 정점

클라우드 시장
확대와 함께
국내 PaaS 시장
성장 필수

PaaS 시장의
성장 가속
시점 도래

국내·외 클라우드 시장 (3/3)

국내 PaaS 시장의 규모는?

2020년 클라우드 서비스별 예상 매출액 (통계청 자료)

구 분	기업수(개)	IaaS	PaaS	SaaS	CSB/CMS	SECaaS	기타	전체	
전체	(1225)	2,378,472	100,091	907,682	352,384	213,276	22,052	3,973,958	
종사자수	1~9인	(312)	201,669	3,254	67,561	6,035	6,397	3,673	288,590
	10~29인	(437)	80,117	38,657	164,925	12,281	44,561	1,565	342,107
	30~99인	(289)	492,598	29,358	241,872	26,836	131,793	10,015	932,472
	100~299인	(121)	223,838	14,395	293,268	12,259	18,597	0	562,357
	300인 이상	(66)	1,380,250	14,427	140,056	294,973	11,928	6,798	1,848,432
기업규모	중견기업 이상	(83)	1,466,305	14,680	196,763	283,480	101,276	6,798	2,069,302
	중소기업	(1142)	912,168	85,412	710,920	68,904	112,000	15,254	1,904,656

전체 3조 9천 7백억 원

IaaS
2조 1천억 원SaaS
9천억 원기타
6천억 원PaaS
1천억 원

2019년 국내 클라우드 서비스별 매출 증가율

1위

SaaS
51.74%
증가

2위

IaaS
22.0%
증가

3위

PaaS
15.71%
증가

2019년 국내 클라우드 서비스별 기업 수 증가율

1위

PaaS
55→131
2017년 대비

2위

SaaS
336→561
2017년 대비

3위

IaaS
314→365
2017년 대비

현재 국내 PaaS 시장 매출규모는 미미하지만,
빠른 시장 성장성을 대비한 기업의 수는 증가하고 있는 상황

클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브는 어떤 특성을 가지고 있는가?



컨테이너 기술 (1/3)

컨테이너 기술을 기반으로 이식성을 제공합니다.

컨테이너, “세계 경제사를 바꾼 大혁신적 발명품” → 수송시간 단축, 물류 수송비 인하



- 세계 화물 운송량 5배 증가
- 화물의 항구 체류 시간 75% 절감
- 해상운송비 60% 절감

형태에나 크기가 다른 물류는 이동 어려움

정형적인 형태의 모양으로 이동에 편리함



컨테이너
혁신적 변화



컨테이너 기술 (2/3)

컨테이너 기술을 기반으로 이식성을 제공합니다.

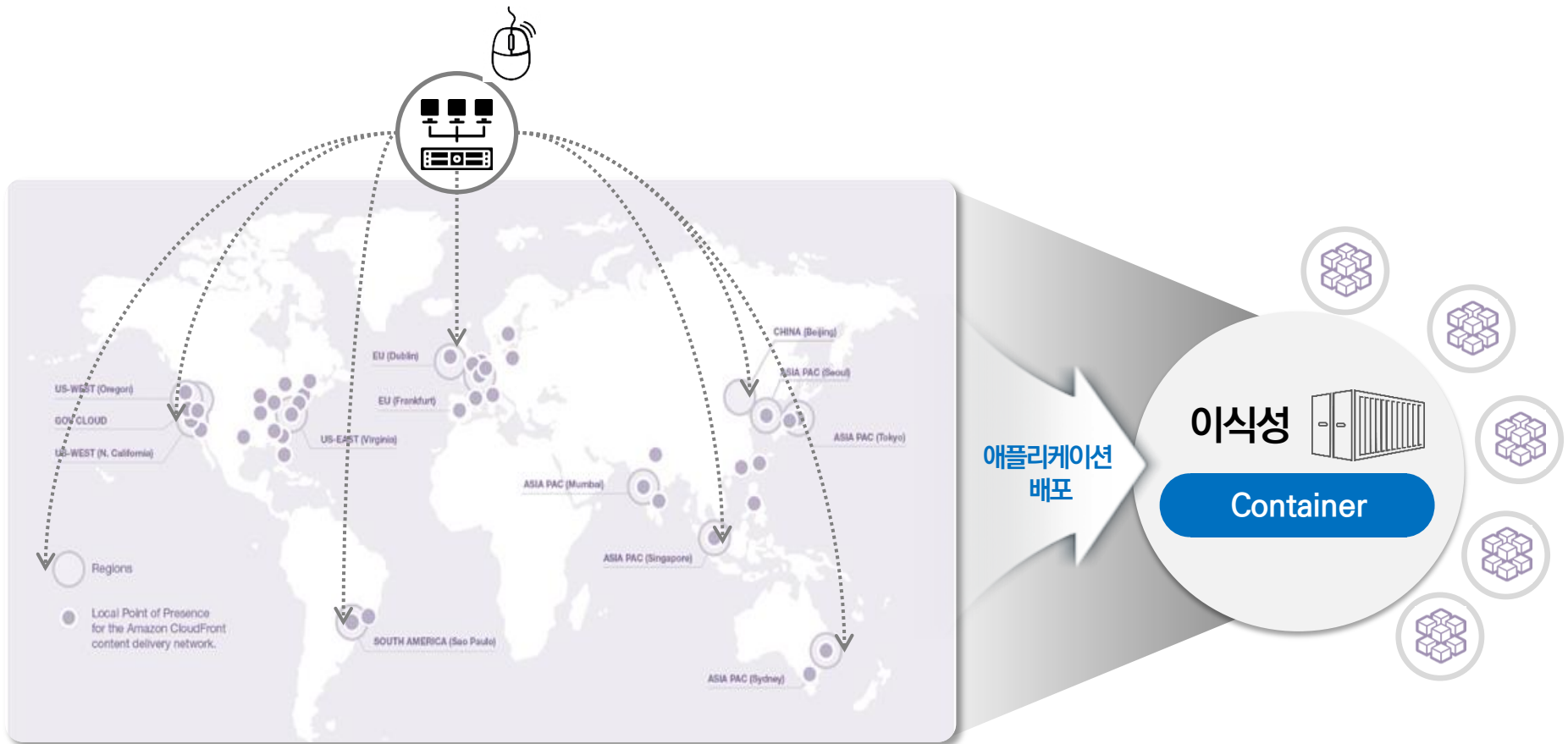
CaaS(Containers-as-a-Service), "세계 IT를 바꾼 大혁신적 발명품" → 서비스 개발 및 운영의 현대화



컨테이너 기술 (3/3)

작고·가볍고 손쉽게 배포를 제공합니다.

클라우드 환경에서의 서비스 배포 → 전세계에 한번의 클릭으로 애플리케이션 배포



클라우드 네이티브 (1/2)

작고·가볍고 손쉽게 배포를 제공합니다.

클라우드 환경에서의 서비스 배포 → 전세계에 한번의 클릭으로 애플리케이션 배포

수 천개 팀 (자율적 DevOps팀)
× 마이크로서비스 아키텍처
× 지속적 배포(CD)
× 다양한 개발 환경

작고/가볍고(Micro) + 손쉽게 배포(DevOps)

= **연간 1억 9천만회 배포**(2020년)
(매초마다 6번의 배포)

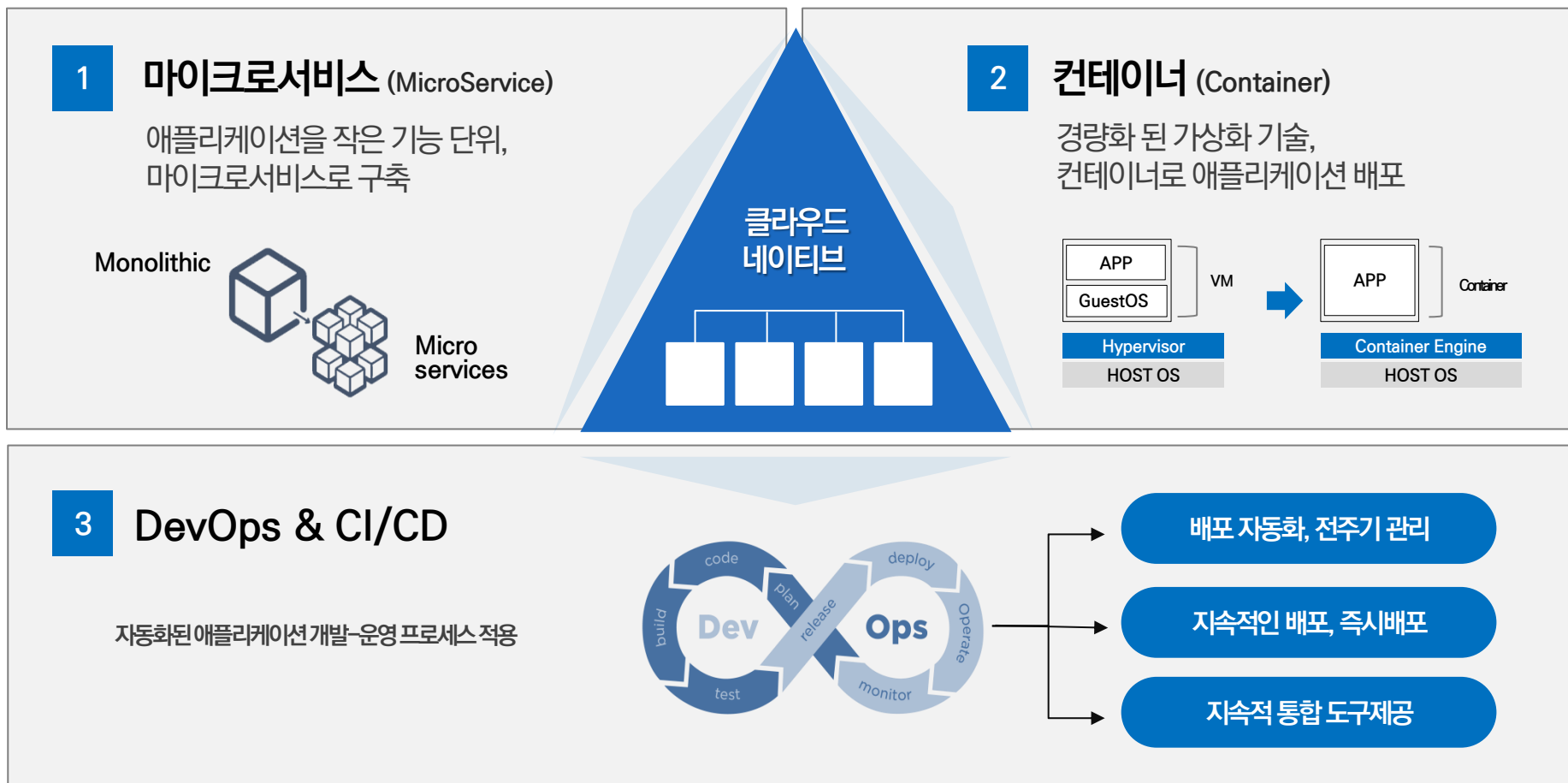
AWS Builders, AWS Fargate와 Amazon ECS를 활용한 CI/CD 모범사례 2020”

aws.amazon.com

클라우드 네이티브 (2/2)

작고·가볍고 손쉽게 배포를 제공합니다.

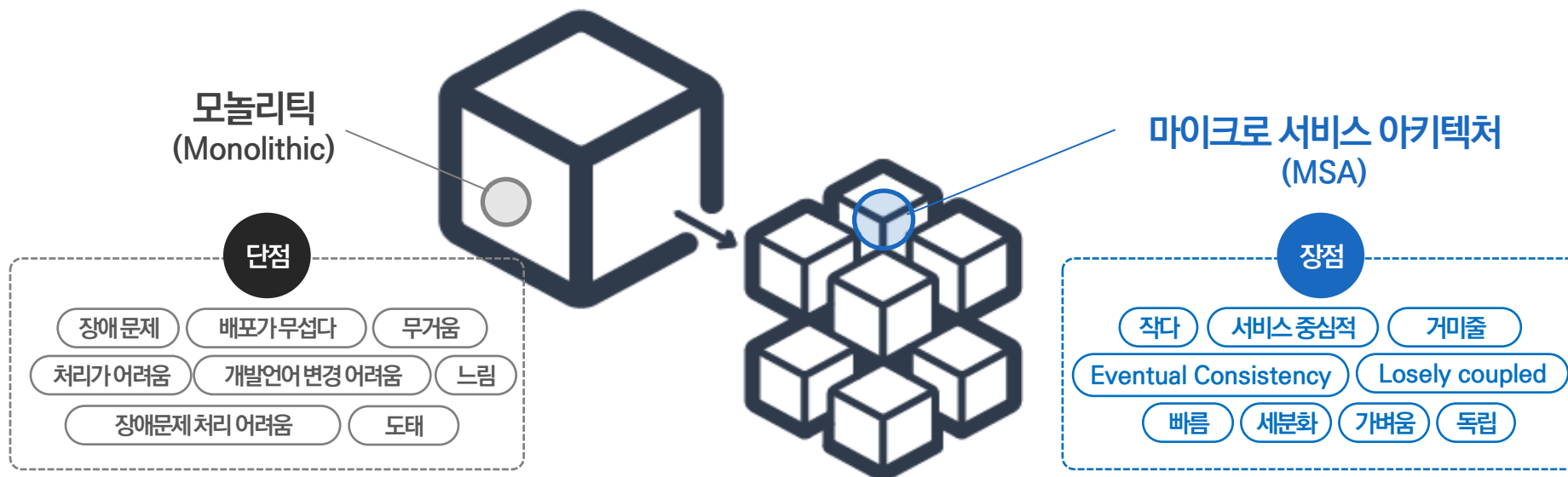
클라우드 환경에서의 서비스 배포 → 전세계에 한번의 클릭으로 애플리케이션 배포



마이크로 서비스 아키텍처 (1/5)

탄력적, 선택적인 서비스 확장을 제공합니다.

API로 통신하는 소규모의 독립적인 서비스로 구성하여,
클라우드 환경에 최적화된 느슨한 결합 (빠른, 신속한, 편리한 업무)

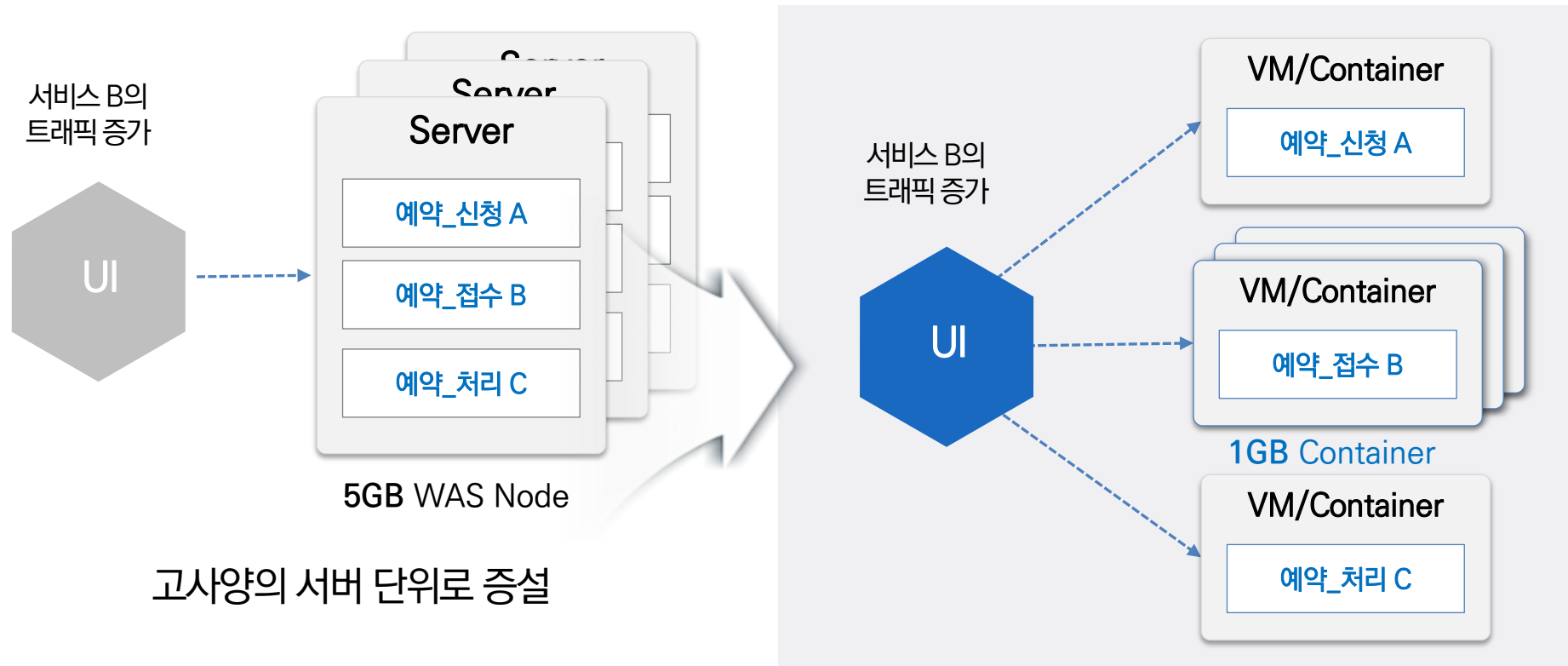


* 마이크로서비스 아키텍처는 통상 MSA로 불리우는데, 클라우드의 특성과 장점을 적용하여, 클라우드 환경에 최적화된 응용SW 설계 기법을 정의

마이크로 서비스 아키텍처 (2/5)

탄력적, 선택적인 서비스 확장을 제공합니다.

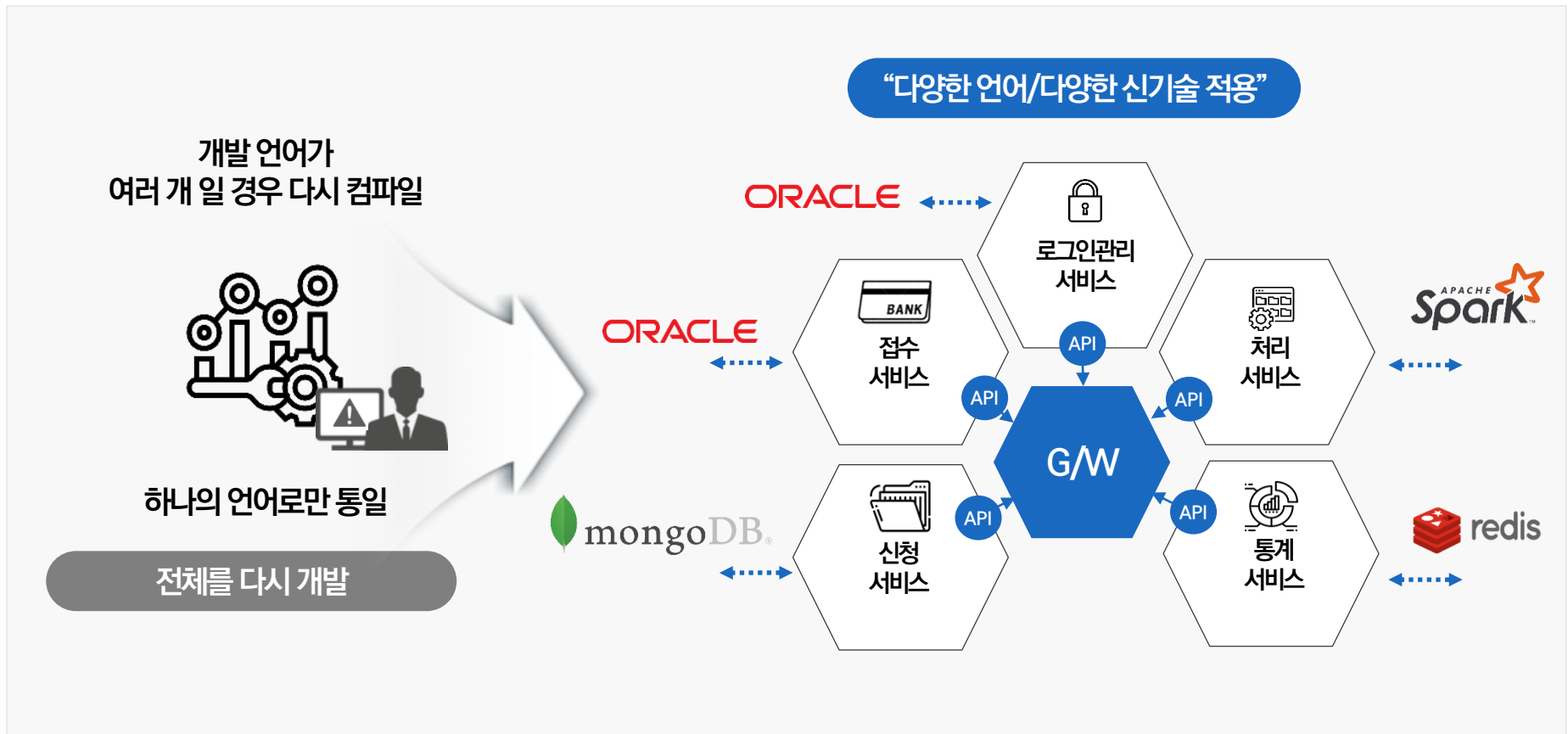
기존에는 요구사항이나 전체를 다시 개발, 빠른 요구에 대응 부합하는 선택적 개발
업무 용도에 적합한 개발언어, DBMS 등의 기술 적용



마이크로 서비스 아키텍처 (3/5)

탄력적, 선택적인 서비스 확장을 제공합니다.

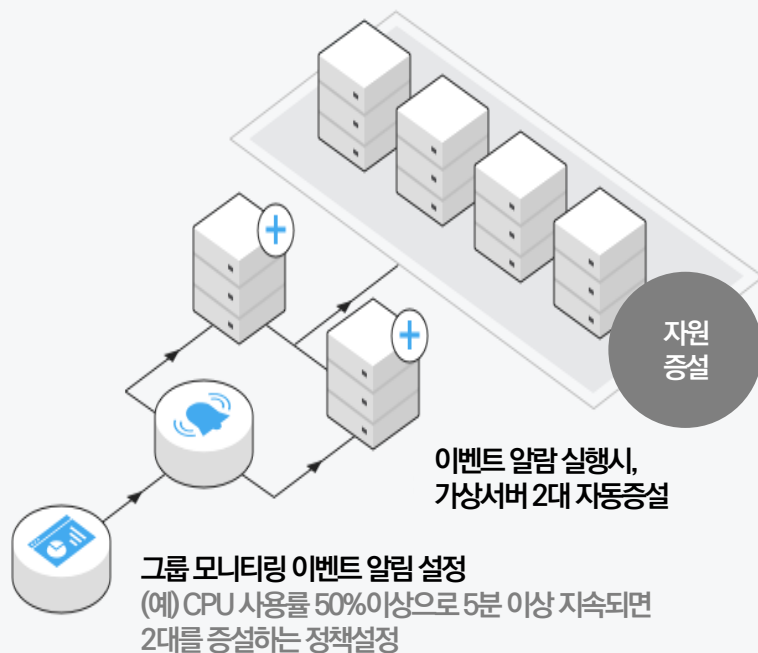
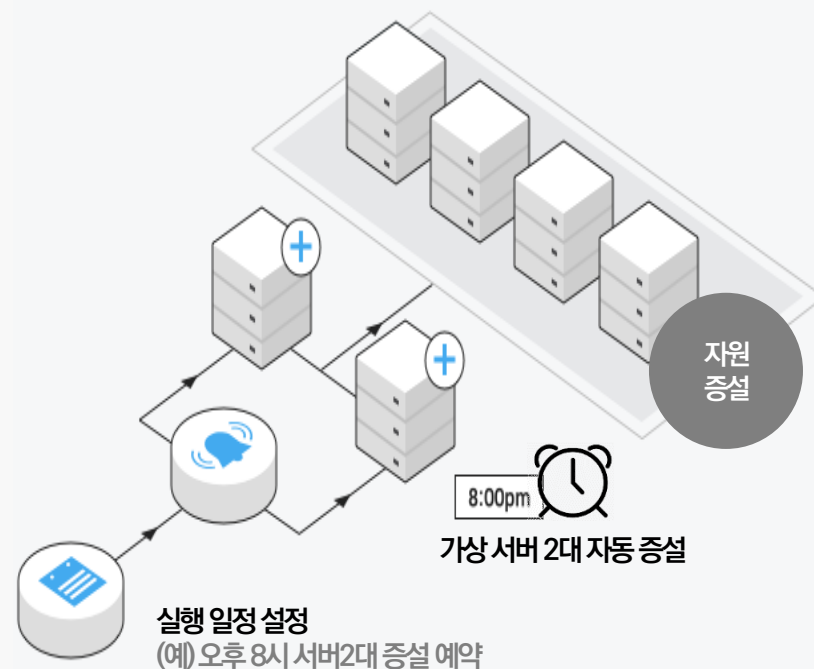
기존에는 요구사항이나 전체를 다시 개발, 빠른 요구에 대응 부합하는 선택적 개발
업무 용도에 적합한 개발언어, DBMS 등의 기술 적용



마이크로 서비스 아키텍처 (4/5)

탄력적, 선택적인 서비스 확장을 제공합니다.

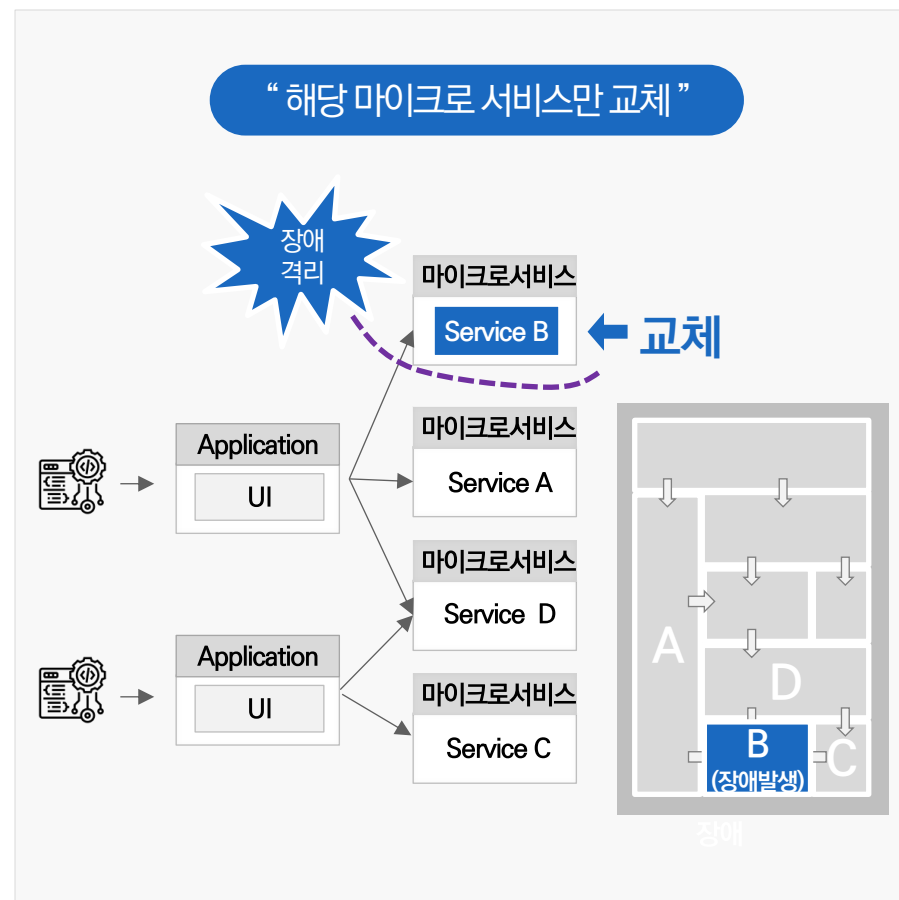
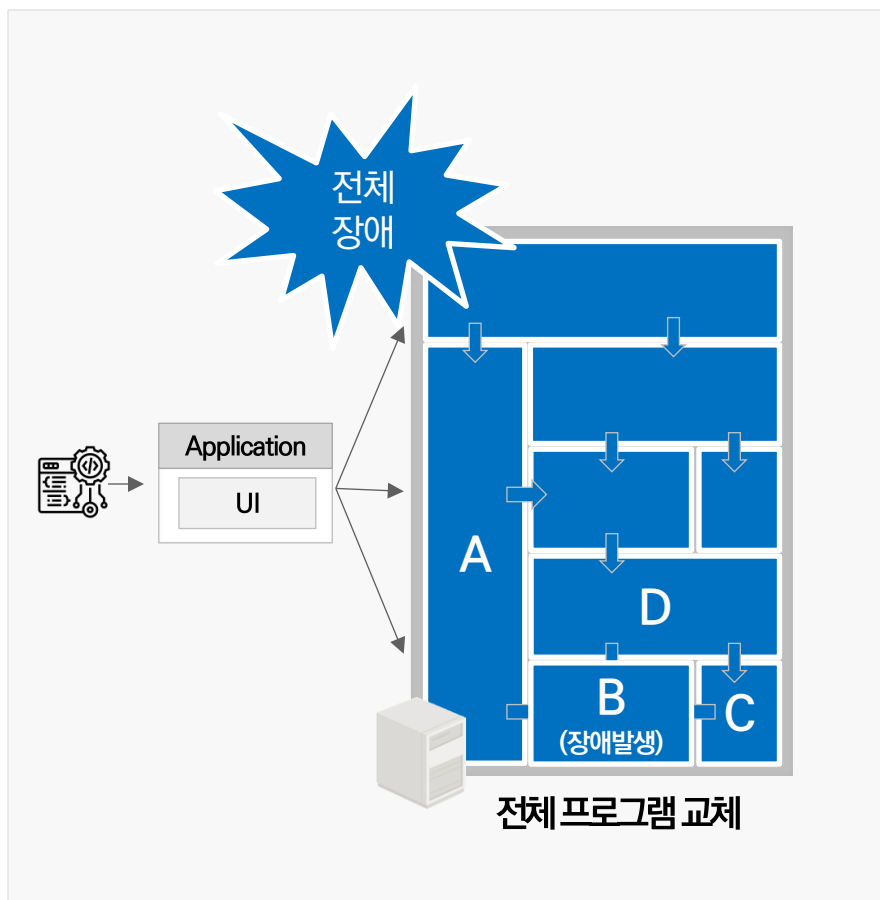
Auto Scaling 시스템 자원들의 메트릭(Metric) 값을 모니터링하여 서버 사이즈를 자동으로 조절하여 업무 **예상치 못한 서비스 부하에 효과적으로 대응**

1 서버 그룹 모니터링 기반 Auto Scaling**2 스케줄링 기반 Auto Scaling**

마이크로 서비스 아키텍처 (5/5)

탄력적, 선택적인 서비스 확장을 제공합니다.

작은 서비스 단위로 업데이트, 변경 및 교체를 제공하며
마이크로 서비스별 격리를 통하여 전체 장애 전파를 최소화



클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브는 어떻게 **발전**하고 있는가?



클라우드 네이티브 정의

클라우드 네이티브란?

클라우드 네이티브
(형용사/명사)

클라우드 컴퓨팅의 장점을 최대한 활용할 수 있는
(효율적인 자원이용, 탄력적 수요 대응 등)
정보시스템 분석·설계·구현 및 **실행하는 환경**

클라우드 네이티브
애플리케이션

클라우드 환경에서 실행되는 **애플리케이션**



CNCF (Cloud Native Computing Foundation) v1.0

클라우드 네이티브 전환할 수 있는 기술 정의 및 오픈 소스를 관리하는 단체

- 퍼블릿, 프라이빗, 하이브리드 **클라우드 환경에서** 확장성 있는 **애플리케이션**
- 컨테이너, 서비스 메시(Mesh), **마이크로서비스(Micro Service) 인프라구조**, 선언적 API로 접근
- 자동화, 회복성, 편리성, 가시성을 갖는 **느슨하게 결합된 시스템** (개발 및 실행 환경)
- 엔지니어는 최소한의 수고로, 영향력이 크고, 예측 가능한 변경을 할 수 있는 기술 정의

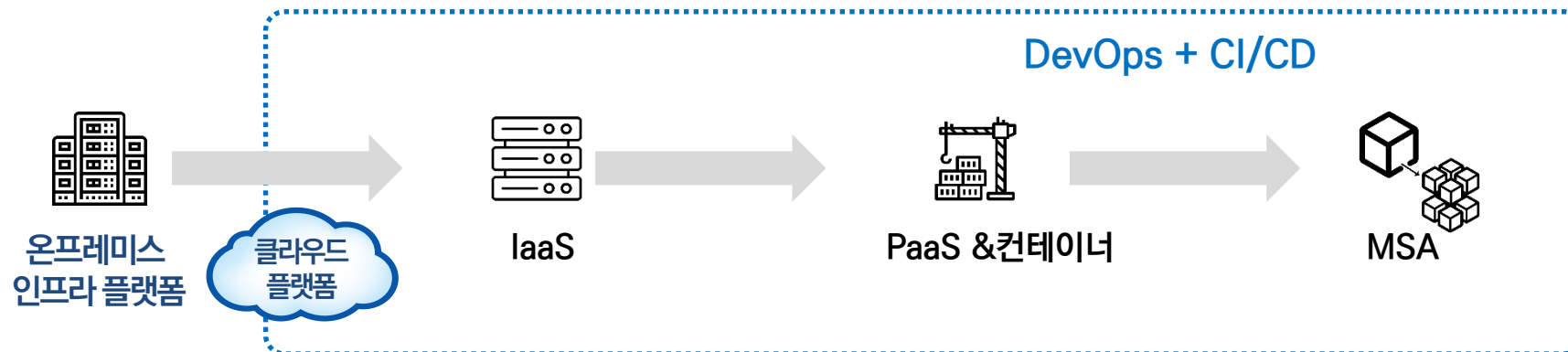
클라우드 네이티브 변화

클라우드 네이티브는 어떻게 발전하고 있는가?



클라우드 네이티브 성숙도 단계 - 애플리케이션 관점

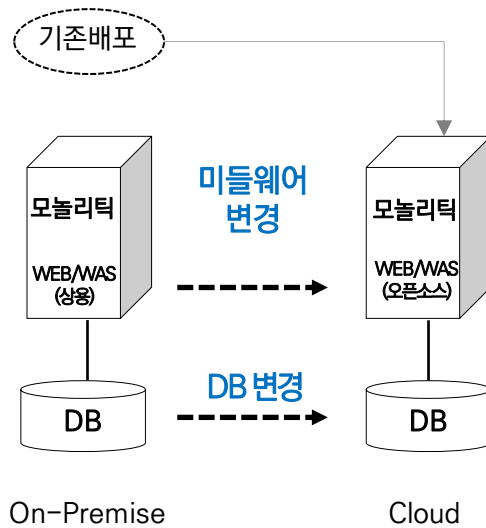
클라우드 애플리케이션



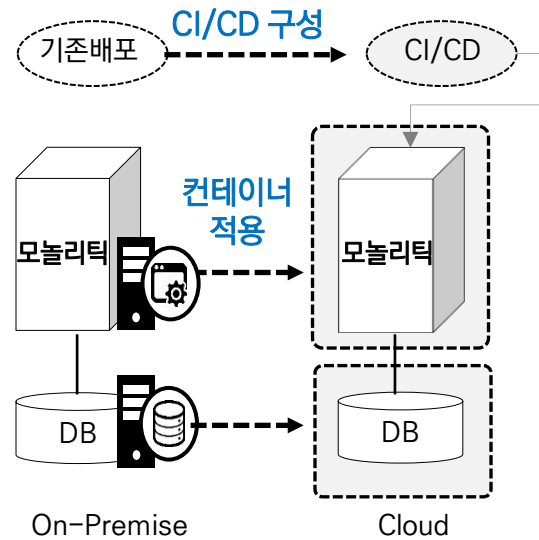
클라우드 네이티브 성숙도 단계 - 인프라 관점

Level 1: Cloud Ready
클라우드 준비 단계

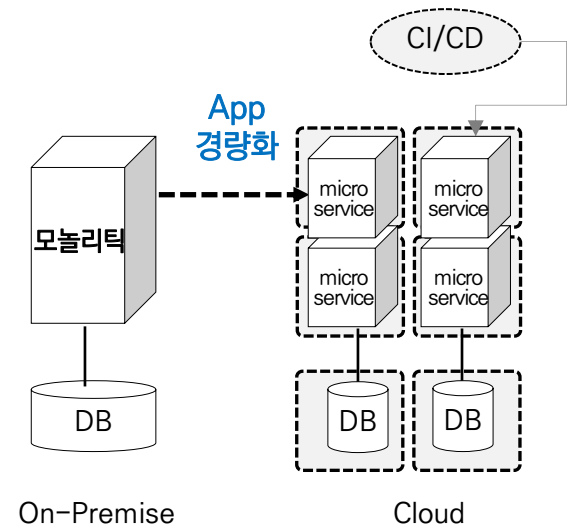
플랫폼 표준화, 비용효율

PaaS
(서버가상화-VM, x86)Level 2: Cloud Friendly
클라우드 친화단계

Auto-Scaling, 컨테이너 적용

PaaS + CI/CD + 12 Factors
(서버가상화-컨테이너, x86)Level 3: Cloud Native
클라우드 네이티브 단계

APP경량화, 업무 민첩성 대응

PaaS + CI/CD + 12 Factors + MSA
(서버가상화-컨테이너, x86)

클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

공공 클라우드와 클라우드 네이티브의 정부정책을 추진합니다.



디지털 정부 정책 변화

디지털 뉴딜, 디지털 정부혁신 및 대국민 수요변화에 능동적으로 대응하고 있습니다.



코로나19 언택트 문화확산



디지털 정부 역할 확대



빠른 보급확산

정부정책



01

한국형 디지털 뉴딜
(DNA생태계 강화)

02

클라우드 산업발전전략
(국가클라우드 대전환)

03

클라우드컴퓨팅 발전계획
(SaaS/PaaS확산)

정부계획



01

민간·공공클라우드 전면전환
(25년 소규모전산실 중심통합)

02

클라우드 플랫폼 고도화
(디지털정부서비스개발환경적용)

03

디지털 서비스
전문계약제도 신설

행정서비스



01

비대면 서비스 요구 증대

02

대국민 디지털 서비스
단기간 제공 급증

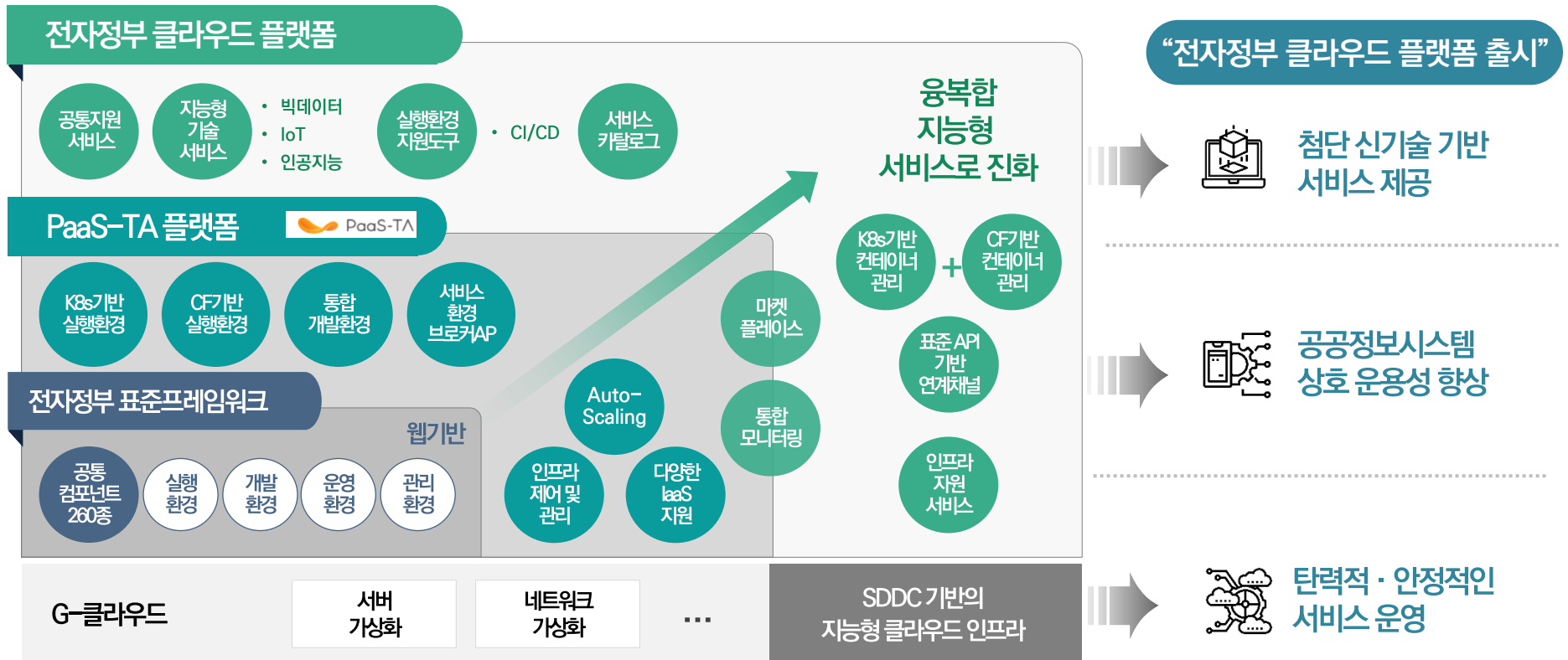
03

전자정부시스템 지원사업
SaaS기반 수출 필요

디지털 정부 변화 (1/2)

'디지털 정부혁신 발전계획'과 '한국판 뉴딜 종합계획'의 일환으로 전자정부 클라우드 플랫폼을 제공합니다.

인프라, 서비스, 개발프레임워크 등을 서비스 형태로 제공하여, 별도의 환경구축 없이 전자정부시스템을 개발·실행·운영·관리할 수 있도록 지원하는 정보화 기반환경으로서 공용활용 플랫폼



디지털 정부 변화 (2/2)

해외는 클라우드 네이티브 전환을 위하여,
선진 정책, 컨설팅과 교육을 제공하고 있습니다.



미국

2019년 6월 기존의
‘클라우드 퍼스트’에서 새로운 전략인
‘클라우드 스마트’로 전환,
효율적인 클라우드 도입 가이드 주력



영국

2017년 2월
클라우드 퍼스트 정책을 재정립
퍼블릭 퍼스트에서 클라우드
네이티브로 전환하도록 장려



싱가포르

IMDA(정보매체개발청)의GoCloud
정책을 통해, ICT중소기업의 개발팀에게
클라우드 네이티브, MSA와 DevOps
관련 컨설팅과 교육제공

출처 : <https://www.imda.gov.sg/programme-listing/gocloud>

평가 도구



- 클라우드 네이티브 진단 도구
를 통한 **기업의 수준 진단**

STEP 01 범위정의



- 클라우드 네이티브 장점 교육
- 전문가와의 협업을 통한
프로젝트 계획 및 범위 정의

STEP 02 교육훈련



- 클라우드 네이티브 애플리케이션
설계, 개발 관련
교육 및 자격증 제공

STEP 03 실습



- 컨설턴트와 고충인력의 클라우드
네이티브 애플리케이션 전환과
클라우드 네이티브 애플리케이션
개발 지원

클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

어떤 공공업무에 클라우드 네이티브의 적용이 가능할까요?



클라우드 네이티브 대상 업무선정 방향 (전문가 의견)

클라우드 네이티브 업무는 학계, 업체,
정부정책을 반영하여 대상을 선정할 수 있습니다.

**서비스 복잡도가 높은 시스템**

- 마틴파울러(2015, 최초 용어정의): “마이크로서비스는 복잡한 시스템에서 유용할 때 MSA전환”

.....

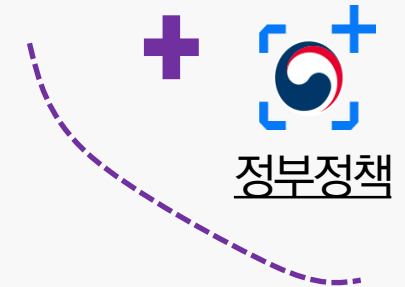
명확한 경계가 가능한 시스템

- 샘뉴먼(2019, 저서): “해당분야를 제대로 이해하지 못해 적절한 경계를 찾기 어렵다면 MSA전환 불리”

.....

더 이상 확장할 수 없는 한계지점에 도달한 시스템

- 수잔파울러(2019, 저서): “확장성 한계로 인해 심각한 안정 문제 발생하여, 개발생산성·효율성 저하 시 MSA 전환”



클라우드 네이티브 적용 검토 (1/3)

클라우드 네이티브는 변화가 많은
대국민 서비스 중심으로 우선 검토하여야 합니다.



정부내 전환가능 업무 식별

✓ SRM(서비스 참조모델)의 전환가능 업무

정부내지원 서비스

업무변화 少
(적용 불리)

공통기술 서비스

시스템 연계 多
(적용 불리)

대국민 서비스

업무변화 多/
폭주성 高
(적용 유리)



기관·시스템특성을 반영한 업무선정

✓ 시스템 특성 반영 (시스템복잡성)



※ 국가 및 기초자치 단체 226, 공공기관 338개 기관 업무대상

✓ 정부정책 특성 반영 (제도개선이 많은 업무)

A기관

B기관

C기관

D기관

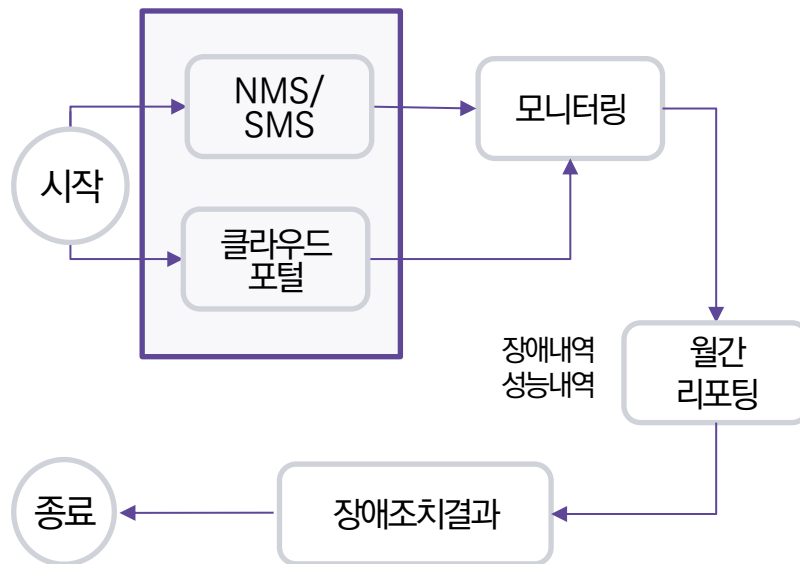
클라우드 네이티브 적용 검토 (2/3)

빈번한 장애, 폭주, 변경 및 배포가 존재하는 한계 지점에 도달한
시스템 중심으로 우선 검토하여야 합니다.



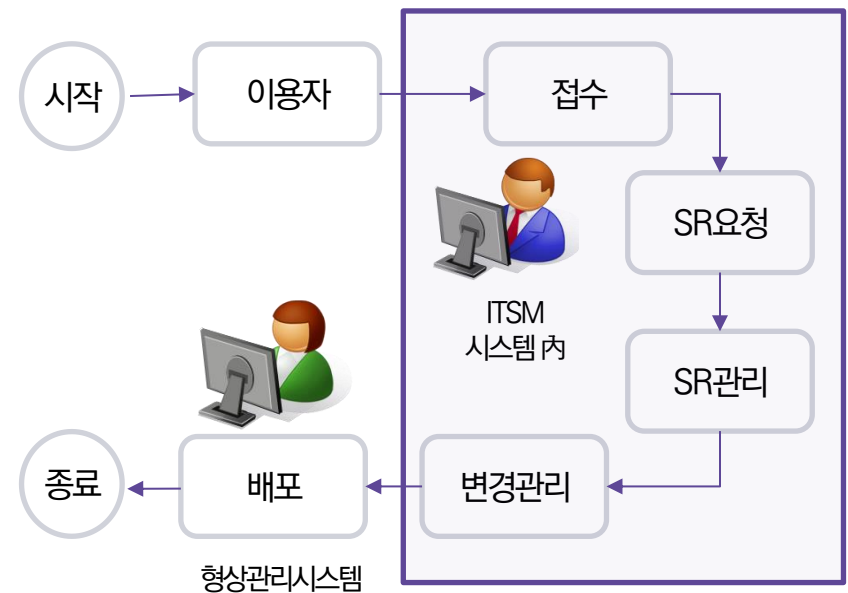
시스템 장애 및 폭주 발생 업무

✓ 시스템 장애 발생내역과 성능분석 절차



빈번한 시스템 변경과 배포 업무

✓ SR 변경신청 및 배포횟수 식별



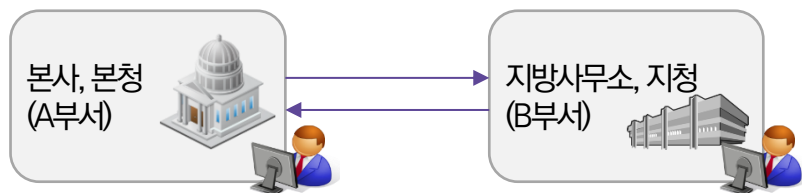
클라우드 네이티브 적용 검토 (3/3)

공공조직의 공통적인 특성(순환보직)을 감안하여,
담당 업무별 기능 단위로 서비스를 분리 구성하여 영향도를 최소화하여야 합니다.



공무원 인사규정 보직변경

☑ 통상적인 조직개편 및 잦은 순환보직



→ 공무원 인사규정:
동일 직위에서 2년 이상 근무한 경우 타 직위 또는 타 부서로 전보

☑ 내 업무 시스템만 관심

타 시스템 (多) - 관련부서 A, B, C, D ~ Z 부서

내 업무시스템(少)

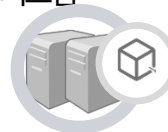
→ 전체 시스템을 모두 이해하기 어려움. 업무담당 시스템 대비 타 시스템 관련(유관부서) 비중이 높음



전체시스템 변경협조는 차세대에서나 가능

☑ 전체 시스템 영향 없이, 내 업무 중심으로 관리하고 배포
(점차적으로 모든 부서가 동일하게 기능 및 서비스단위 분리)

내 업무시스템
변경

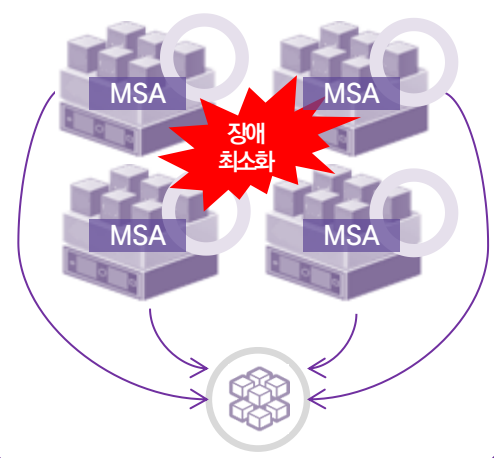


어떤 영향,
장애 걱정



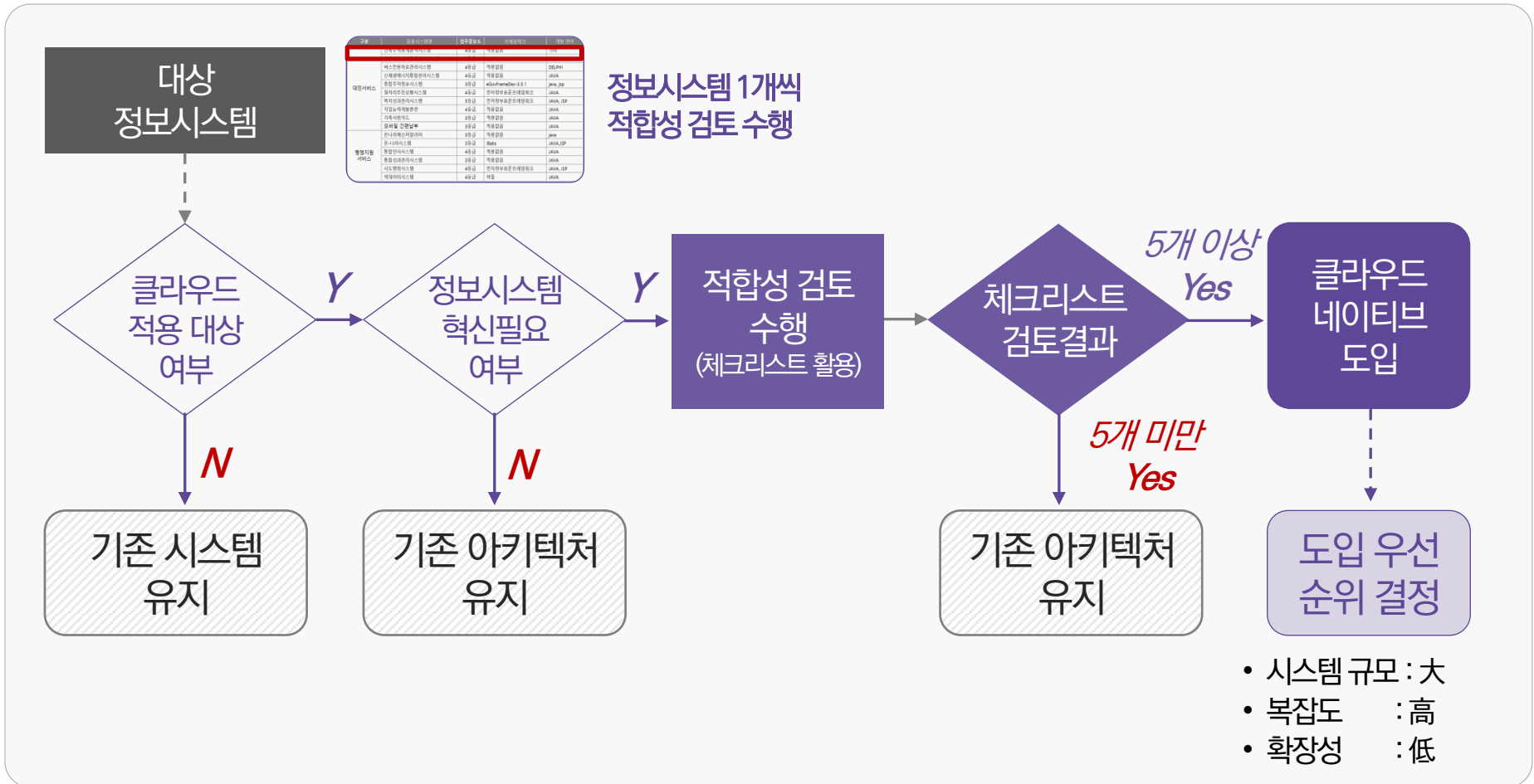
'내 프로그램 변경에
타 시스템이
영향 없도록 ...'

전체 시스템 영향 거의 없도록 하여야 함



정보시스템 자가진단 방법

클라우드 네이티브 구성요소의 특징점을 토대로,
현행 정보시스템에 대한 체크리스트를 도출 하였습니다.



정보시스템 자가진단 (1/3)

현행 정보시스템에 대한 체크리스트를 도출하였으며, 5개 이상 “Y” 응답시,
클라우드 네이티브 도입이 필요한 것으로 판단할 수 있습니다.



■ 정보시스템 자가진단 체크리스트 예시

구분	적합성 검토 항목	자가 진단 항목	답변 (예/아니오)
1. 정책 및 업무 변화 대응	정책 및 업무 변화 시 민첩한 대응	1 1-1. 한국판 뉴딜 등 새로운 정책, 기관별 업무계획 등에 따른 다양한 정보화 요구사항 변화에 대해 신속한 대응이 필요합니까?	5개 이상 “YES” 응답 시 도입 검토
	디지털 혁신 및 지능화 지원	2 1-1. 지능 정보화 관련 디지털 신기술(빅데이터, AI, 블록체인, IoT 등) 도입 관련 요구사항 반영이 필요합니까?	
2. 안정적 서비스 운영	서비스 개선 요구사항 증가	3 2-1. 서비스 이용자의 잦은 서비스 개선 요구(CSR)에 대해 적시 대응이 필요하거나 초기개발비의 약 15% 이상을 매년 추가개발 및 유지보수 비용으로 사용하고 있습니까?	
	접속 지연, 서비스 장애 문제 신속한 해결	4 2-2. 다양한 원인에 의한 장애 발생 시 장애복구(예.시스템 증설, 업그레이드 등)를 위해 서비스를 중단한 적이 있거나 특정시점(년, 월, 주, 시)에 트래픽의 증가에 의한 접속지연으로 불만이 제기된 적이 있습니까?	
	소규모 서비스 분리 및 독립적 운영	5 2-3. 소규모 서비스 단위로 기능과 DB의 명확한 분리가 가능하고, 독립적 단위로 실행이 가능합니까? (공통 기능 및 데이터 사용 유무, 타 시스템과의 연계성, 서비스 의존관계 등 확인)	
	다양한 플랫폼 환경에서의 이식성 보장	6 2-4. 개발-테스트-운영 환경의 일치를 통한 배포가 필요하거나 다수의 플랫폼에 애플리케이션 배포가 필요합니까?	
3. 개발 품질 향상	개발-운영 협업 조직체계 구현	7 3-1. 시스템 개발 및 운영 시 개발 및 운영 조직의 분리에 따라 의사소통, 개발 및 배포 지연 등의 문제가 존재합니까?	
	전문인력 역량 강화	8 3-2. 배포 관련 개발 품질 향상을 위해 전문 인력 확보와 인력의 역량 강화가 필요합니까? (단, 개발-운영 인력이 1-2명 존재하는 경우는 “해당 없음”)	
4. 개발 생산성 향상	코딩-빌드-테스트-배포 파이프라인 자동화	9 4-1. 개발된 SW를 형상관리 시스템에 커밋한 후 개발, 검증, 운영 서버에서 빌드, 테스트, 배포하는 과정에 빌드·테스트·배포 도구를 사용하지 않거나 부분적으로 사용하고 있습니까?	
	기술 반영 용이성 확보	10 4-2. 오픈소스 SW 등 다양한 기술의 도입, 업그레이드, 업데이트 등 작업 수행 시 연관된 서비스에 대한 변경 작업이 복잡하여 개발의 어려움이 존재합니까?	

정보시스템 자가진단 (2/3)

“YES” 답변이 3개 항목 이하라면,
기존 아키텍처를 유지를 권장합니다.



표준 프레임워크 포털 적용 예시

*자가 진단 항목을 활용하여 체크함

구분(목표)	자가 진단 항목	근거	검토결과
1. 정책 및 업무 변화 대응	1 1-1. 한국판 뉴딜 등 새로운 정책, 기관별 업무계획 등에 따른 다양한 정보화 요구사항 변화에 대해 신속한 대응이 필요합니까?	• 해당 없음	
	2 1-1. 지능 정보화 관련 디지털 신기술(빅데이터, AI, 블록체인, IoT 등) 도입 관련 요구사항 반영이 필요합니까?	• 해당 없음	
2. 안정적 서비스 운영	3 2-1. 서비스 이용자의 잦은 서비스 개선 요구(CSR)에 대해 적시 대응이 필요하거나 초기개발비의 약 15% 이상을 매년 추가개발 및 유지보수 비용으로 사용하고 있습니까?	• 행안부와 NIA의 요구사항 발생 시 (보안 점검, 호환성 체크 등)	✓
	4 2-2. 다양한 원인에 의한 장애 발생 시 장애복구(예:시스템 중설, 업그레이드 등)를 위해 서비스를 중단한 적이 있거나 특정시점(년, 월, 주, 시)에 트래픽의 증가에 의한 접속지연으로 불만이 제기된 적이 있습니까?	• 장애 발생하지 않음 (G-클라우드 구성 관련 서비스 중단 시 사전 공지한 적 있음)	
	5 2-3. 소규모 서비스 단위로 기능과 DB의 명확한 분리가 가능하고, 독립적 단위로 실행이 가능합니까? (공통 기능 및 데이터 사용 유무, 타 시스템과의 연계성, 서비스 의존관계 등 확인)	• 서비스 단위로 기능과 서비스 분리 가능	✓
	6 2-4. 개발-테스트-운영 환경의 일치를 통한 배포가 필요하거나 다수의 플랫폼에 애플리케이션 배포가 필요합니까?	• 개발-테스트-운영 환경에 배포하지만 타 플랫폼 이식은 없음	
3. 개발 품질 향상	7 3-1. 시스템 개발 및 운영 시 개발 및 운영 조직의 분리에 따라 의사소통, 개발 및 배포 지연 등의 문제가 존재합니까?	• 개발자 1명 (유지보수 10%, 운영 90%)	
	8 3-2. 배포 관련 개발 품질 향상을 위해 전문 인력 확보와 인력의 역량 강화가 필요합니까? (단, 개발-운영 인력이 1-2명 존재하는 경우는 “해당 없음”)	• 해당 없음	
4. 개발 생산성 향상	9 4-1. 개발된 SW를 형상관리 시스템에 커밋한 후 개발계, 검증계, 운영계 서버에서 빌드, 테스트, 배포하는 과정에 빌드·테스트·배포 도구를 사용하지 않거나 부분적 ¹⁾ 으로 사용하고 있습니까?	• G-클라우드에서 제공하지 않음(KT 내부 정책) - Jenkins 등 지원하지 않음 ※ 자체적으로 Jira, SVN, Git·hub, Git·Lab, Jenkins 사용	✓
	10 4-2. 오픈소스 SW 등 다양한 기술의 도입, 업그레이드, 업데이트 등 작업 수행 시 연관된 서비스에 대한 변경 작업이 복잡하여 개발의 어려움이 존재합니까?		

정보시스템 자가진단 (3/3)

안정적 서비스운영, 업무·기술변화대응, 개발품질 향상, 개발기간 단축이라는 목표를 달성할 수 있습니다.



공영홈쇼핑 적용 예시

*자가 진단 항목을 활용하여 체크함

구분(목표)	질의사항	근거	검토결과
1. 정책 및 업무 변화 대응	1 1-1. 한국판 뉴딜 등 새로운 정책, 기관별 업무계획 등에 따른 다양한 정보화 요구사항 변화에 대해 신속한 대응이 필요합니까?	• 매년 업무계획에 따른 정보화 요구사항 반영 필요	✓
	2 1-1. 지능 정보화 관련 디지털 신기술(빅데이터, AI, 블록체인, IoT 등) 도입 관련 요구사항 반영이 필요합니까?	• 해당 없음	
2. 안정적 서비스 운영	3 2-1. 서비스 이용자의 잦은 서비스 개선 요구(CSR)에 대해 적시 대응이 필요하거나 초기개발비의 약 15% 이상을 매년 추가개발 및 유지보수 비용으로 사용하고 있습니까?	• 홈쇼핑 특성상 필요 - 11번가, CJ홈쇼핑 등과 유사	✓
	4 2-2. 다양한 원인에 의한 장애 발생 시 장애복구(예.시스템 증설, 업그레이드 등)를 위해 서비스를 중단한 적이 있거나 특정시점(년, 월, 주, 시)에 트래픽의 증가에 의한 접속지연으로 불만이 제기된 적이 있습니까?	• 가용성 99.5%	
	5 2-3. 소규모 서비스 단위로 기능과 DB의 명확한 분리가 가능하고, 독립적 단위로 실행이 가능합니까? (공통 기능 및 데이터 사용 유무, 타 시스템과의 연계성, 서비스 의존관계등 확인)	• 서비스별 기능 및 DB 분리 가능	✓
	6 2-4. 개발-테스트-운영 환경의 일치를 통한 배포가 필요하거나 다수의 플랫폼에 애플리케이션 배포가 필요합니까?	• 개발-테스트-운영 환경에 배포하지만 다 플랫폼 이식은 없음	
3. 개발 품질 향상	7 3-1. 시스템 개발 및 운영 시 개발 및 운영 조직의 분리에 따라 의사소통, 개발 및 배포 지연 등의 문제가 존재합니까?	• 정기/수시 배포 중, 절차적으로 문제 없도록 처리 - 단계별 승인 후 진행	
	8 3-2. 배포 관련 개발 품질 향상을 위해 전문 인력 확보와 인력의 역량 강화가 필요합니까? (단, 개발-운영 인력이 1-2명 존재하는 경우는 “해당 없음”)	• 현황 조사 중	
4. 개발 생산성 향상	9 4-1. 개발된 SW를 형상관리 시스템에 커밋한 후 개발계, 검증계, 운영계 서버에서 빌드, 테스트, 배포하는 과정에 빌드·테스트·배포 도구를 사용하지 않거나 부분적 ¹⁾ 으로 사용하고 있습니까?	• 홈쇼핑 특성상 수시 상품 변경에 따른 신속한 배포 필요 - 빌드, 배포 도구 일부 사용	✓
	10 4-2. 오픈소스 SW 등 다양한 기술의 도입, 업그레이드, 업데이트 등 작업 수행 시 연관된 서비스에 대한 변경 작업이 복잡하여 개발의 어려움이 존재합니까?	• 24x365 서비스 제공으로 각종 업그레이드 작업 시 서비스 제공의 어려움 존재	✓

5개
항목에
“예” 응답

클라우드
네이티브
도입 필요

(컨테이너,
DevOps &
CI/CD, MSA
도입 검토)

클라우드 네이티브 업무 도입 검토사항

클라우드 네이티브를 도입시 사전 고려사항을 검토합니다.



장기적 비즈니스 목적
(명확한 기능경계, 시스템 확장)



MSA에 필요한 기술 보유



비즈니스 도달까지 충분한 시간
(다양한 신기술 적용을 위한 충분한 개발기간)



요구분석, 설계, 개발 테스트
출시를 반복적으로 수행

서비스 조직구성과 Agile 문화
(개발~운영을 합친 계약방식 도입 검토)



**비즈니스 규모에 맞는
충분한 인력 사전 배치**



개발자의 숙련도 및 도구 선정
(기술적 숙련도가 높은 개발자의 참여)

클라우드 네이티브 애플리케이션(MSA) 도입 조건

1. 빠르고 잦은 배포가 필요한가?

도입
조건
1

MSA의 근본적인 목적은 빠른 비즈니스 대응, 민첩성에 있음
얼마나 사용자의 요구사항을 빠르게 반영하고 개선해 나갈 수 있을지에 대해 초점이 맞추어져 있음
자주 개선해야 되거나 배포해야 되는 시스템(서비스)이 아닌 경우는 MSA가 적합하지 않을 수 있음

2. 성능이 중요한 서비스 인가?

도입
조건
2

분리된 서비스들은 물리적인 네트워크 통신을 하게 되므로 네트워크 대기시간(Latency)이 발생할 수 있음
모놀리스에 비해 성능이 저하될 수 있으므로 성능에 민감한 시스템(서비스)인 경우 도입을 고려해야 함

3. 분산 트랜잭션이 필요한 서비스 인가?

도입
조건
3

모놀리스에서 여러개의 서비스들을 묶어 단일트랜잭션으로 처리할 수 있었다면, MSA에서는 단일 트랜잭션 처리가 어려움
이벤트 기반의 트랜잭션을 구현하여 데이터의 정합성을 유지할 수 있음

4. 모놀리식에 비해 비용을 절감할 수 있는가? (비용대비 이득을 따져 볼 것)

도입
조건
4

서비스들이 늘어난다는 것은 관리 할 서버가 늘어난다는 것을 의미함. 그러므로 운영관리 비용이 증가할 수 있음
하지만 MSA가 주는 장점의 이득이 더 크다고 판단될 경우, MSA를 도입하는 것이 좋은 선택이 될 수 있음

5. 조직의 개발 문화는 준비되어 있는가?

도입
조건
5

단순히 아키텍처만 도입한다고 해서 MSA가 구현되는 것은 아니며, MSA에 적합한 조직의 개발문화(DevOps)를 정착시킬수 있도록 노력해야 함

클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브 도입 고려사항은 어떻게 되나요?



정보화 사업관리 프로세스

공공기관에서 추진하는 정보화 사업관리는 기획·검토, 계획수립, 사업자 선정계약, 개발구축, 감리, 검사종료 단계에 따라 진행됩니다.



클라우드 네이티브 추진 관련 사항 검토 대상

출처: 행정기관을 위한 정보화사업 단계별 관리점검 가이드 V2.0, 행정안전부 & NIA

사업발주를 위한 사업관리 단계별 고려사항 (1/2)

기획·검토, 계획수립, 사업자선정·계약, 개발·구축, 감리, 검사 종료, 사업자 선정·계약 이슈입니다.

발주 단계			클라우드 네이티브 사업단계별 고려사항
기획·검토	기획·검토	정보화사업 기획	<ul style="list-style-type: none"> 클라우드 네이티브 시스템 구축 관련 성과계획 수립
계획수립	사업계획서(안) 작성	사업계획서(안) 작성	<ul style="list-style-type: none"> 클라우드 네이티브 관련 요구사항 상세화 클라우드 네이티브 도입을 고려한 개발비 예산 산정
		기술적용계획 수립	<ul style="list-style-type: none"> 클라우드 네이티브 애플리케이션 관련 기술적용계획 수립
	사업계획서(안) 검토	기술평가 시행	<ul style="list-style-type: none"> 클라우드 네이티브 관련 요구사항 상세화
사업자 선정·계약	제안 요청	제안요청서 작성	<ul style="list-style-type: none"> 클라우드 네이티브 관련 요구사항 상세화 클라우드 네이티브 도입을 고려한 개발비 예산 반영

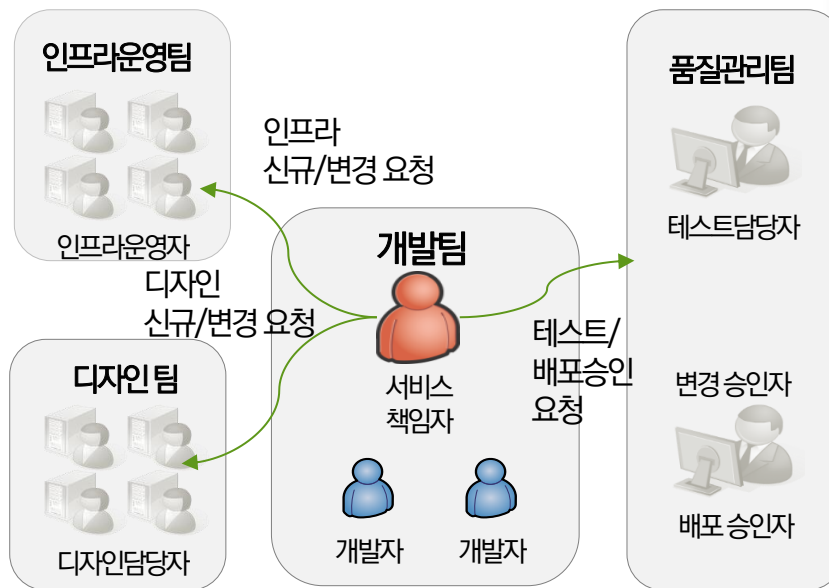
사업발주를 위한 사업관리 단계별 고려사항 (2/2)

발주 단계			클라우드 네이티브 사업단계별 고려사항
사업자 선정·계약	사업자 선정 및 계약 체결	제안서 평가	• 기술평가수행 시 클라우드 네이티브 적용 경험 및 기술력을 평가할 수 있는 항목 정의 및 배점 부여
		협상 및 낙찰자 선정	• 기술협상 시 클라우드 네이티브 관련 기술적 이행 사항에 대해 구체적으로 협상 수행
개발·구축	사업착수	착수계획서 검토	• 착수계획서 내에 클라우드 네이티브 관련 과업 내용이 누락 없이 잘 반영되어 있는지 확인
	진도/품질관리	진도/품질관리	<ul style="list-style-type: none"> • 방법론의 공정단계별 품질관리 활동 수행 <ul style="list-style-type: none"> - 개발방법론의 공정단계별 품질관리 활동 수행 - 클라우드 네이티브 관련 요구사항별 이행 과정 점검
감리	감리시행	감리시행	<ul style="list-style-type: none"> • 감리계획서 및 수행결과보고서 검토 및 조치 <ul style="list-style-type: none"> - 감리계획서 및 수행결과보고서 내에 클라우드 네이티브 관련 요구사항에 대한 점검계획과 결과 검토 후 조치 요구
검사종료	완료검사	완료검사	• 완료검사수행 시 클라우드 네이티브 관련 요구사항의 이행 여부 확인

클라우드 네이티브 조직 변화

인프라·플랫폼/디자인/개발/운영 비즈니스에 민첩한 대응이 가능한 조직으로 변화해야 합니다.

일반적 개발·운영 방식



개발팀이 인프라/디자인/품질관리팀에 연락하여
신규·변경을 요청

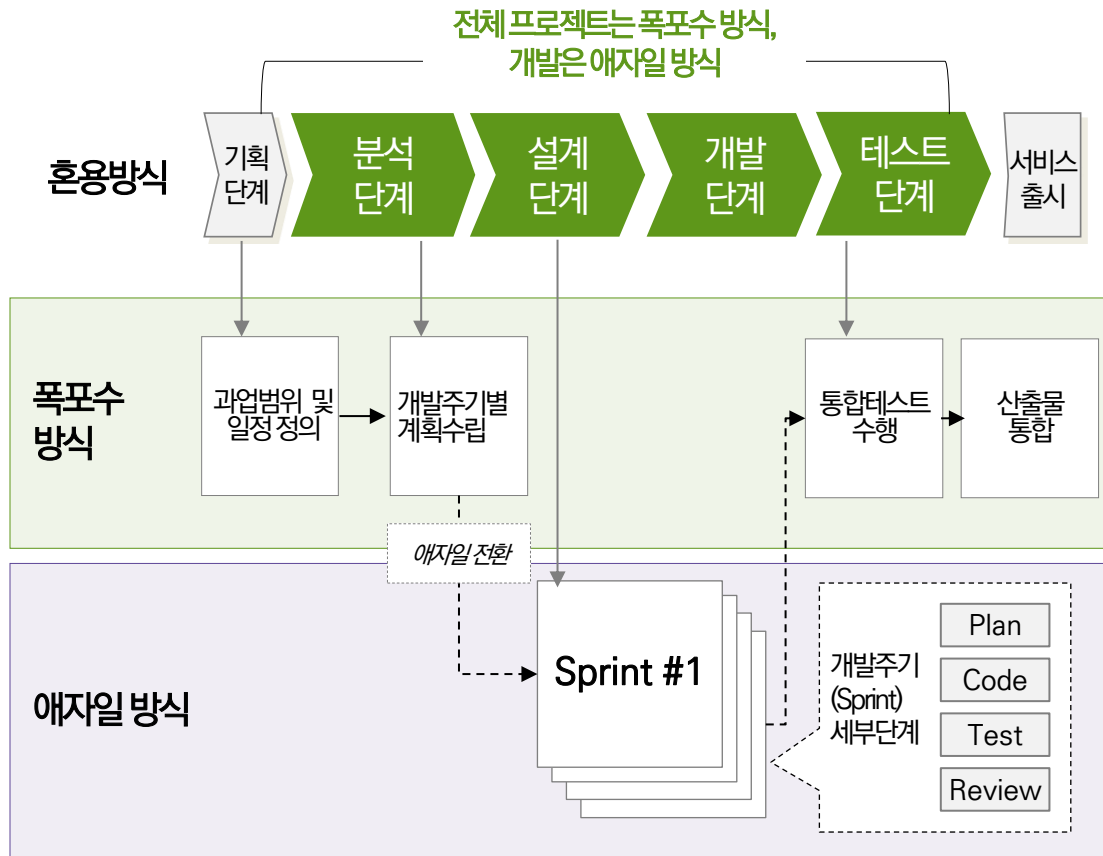
마이크로서비스 제공을 위한 개발·운영 방식



서비스 단위로 DevOps조직을 운영하고
운영환경에 직접배포

클라우드 네이티브 환경조성을 위한 애자일 방법론 적용

공공부문 애자일 적용은 현실적으로 애자일 방법론을 폭포수 방법론의 보완적인 역할로 폭포수-애자일 혼용방식(하이브리드)의 적용이 합리적입니다.



계약방식	<ul style="list-style-type: none"> • 확정된 예산금액 만큼 고정가 계약 • 과업범위, 수행인력 및 계약기간 명시
과업범위	<ul style="list-style-type: none"> • 요구사항 및 품질기준을 최대한 확정 후 프로젝트 착수
개발방식	<ul style="list-style-type: none"> • 개략적인 개발일정으로 프로젝트 착수, 개발주기(Sprint)별 상세계획에 따라 설계, 개발, 테스트 및 평가 수행
결과물 (산출물)	<ul style="list-style-type: none"> • 개발주기(Sprint)별 산출물 분할 작성 후 프로젝트 종료시점까지 산출물 통합 및 검수 진행
수행기간	<ul style="list-style-type: none"> • 계약서상의 도급계약 기간 준수

※ 국내 공공부문의 SW 조달체계는 기본적으로 과업범위(요구사항 확정)와 예산이 확정된 고정가 계약을 전제로 하고 있어, 계약에 명시된 목적물(시스템 및 관련 산출물)을 정해진 기간안에 제공. 국외 공공부문의 경우 애자일을 도입하는 사례가 점점 증가하고 있음

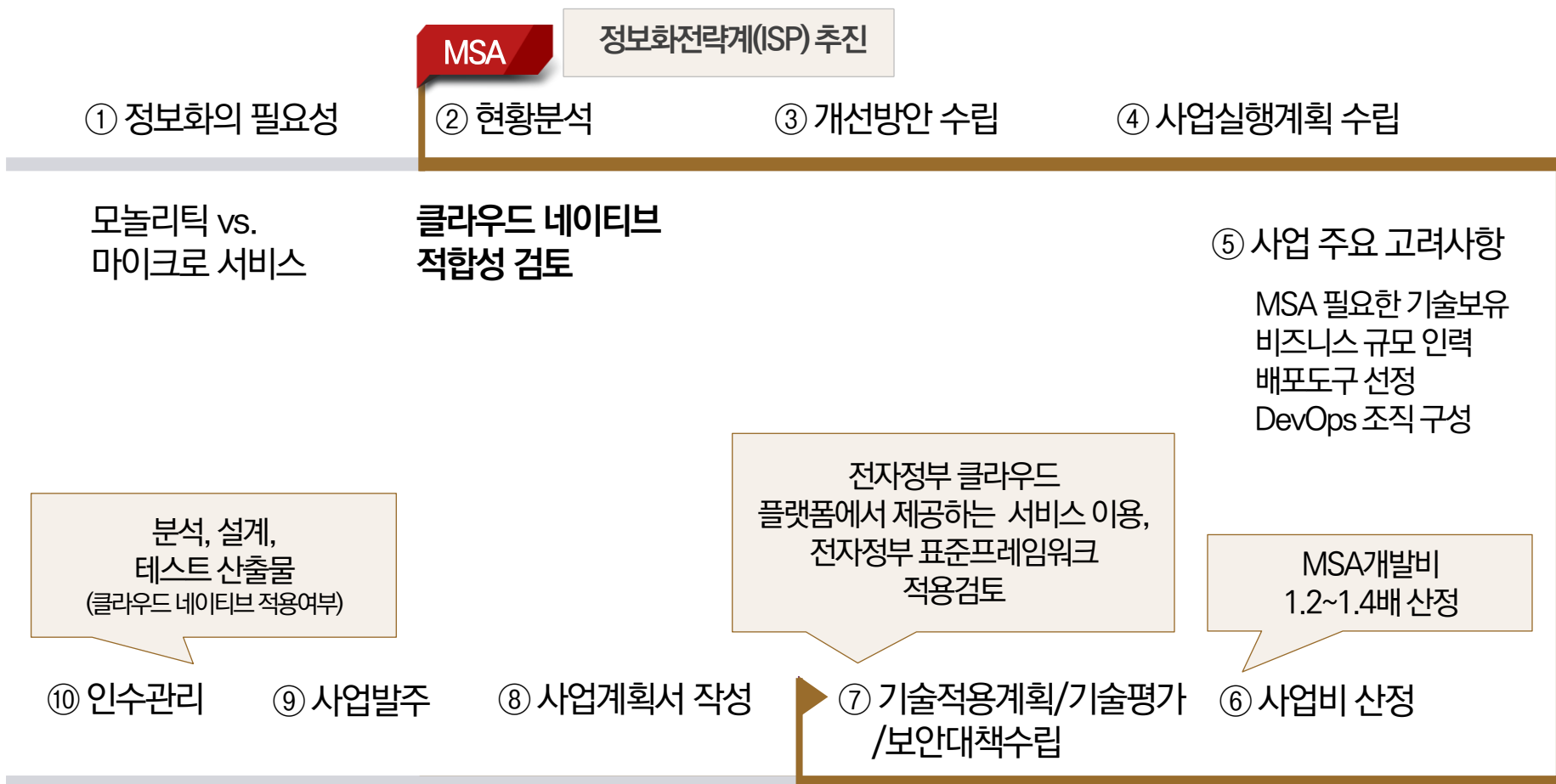
클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브 도입 절차는 어떻게 되나요?



클라우드 네이티브 프로세스

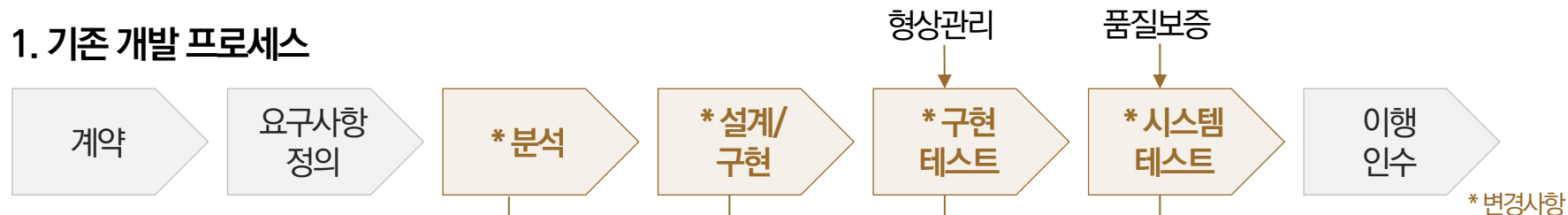
행정기관을 위한 정보화 사업 단계별 관리 및 점검가이드를 참조하여 수립한
클라우드 네이티브 발주자(기획자) 프로세스입니다.



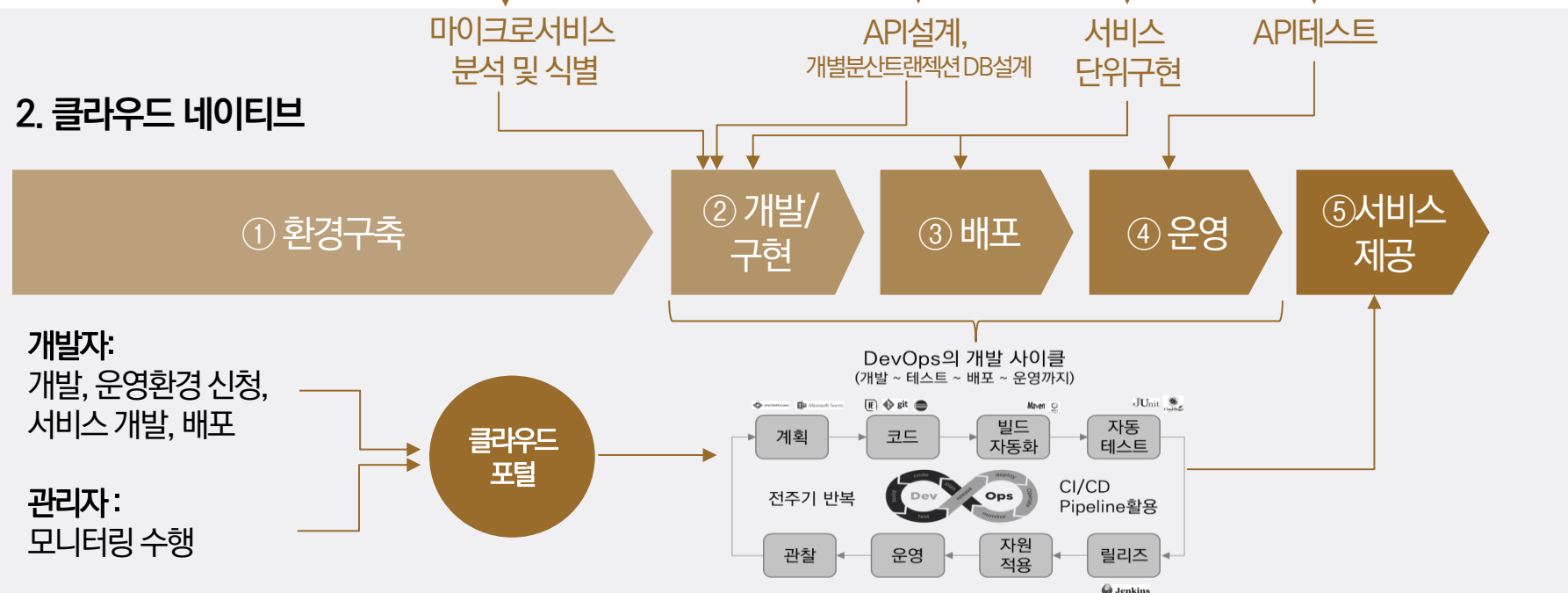
클라우드 네이티브 프로세스

행정기관을 위한 정보화 사업 단계별 관리 및 점검가이드를 참조하여 수립한 클라우드 네이티브 개발자 프로세스입니다.

1. 기존 개발 프로세스



2. 클라우드 네이티브

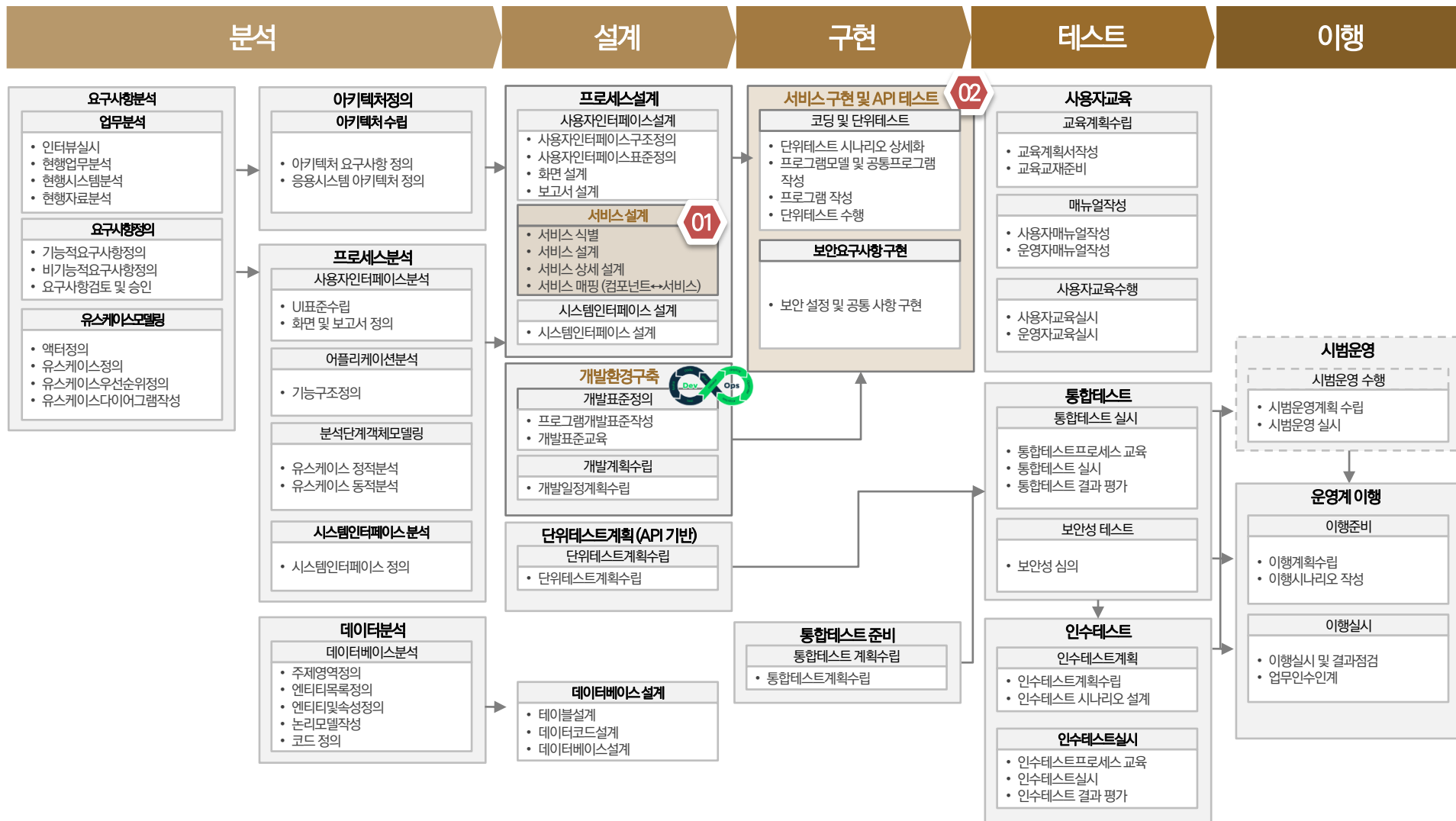


전자정부 클라우드 플랫폼 기반 클라우드 네이티브 애플리케이션 개발 절차



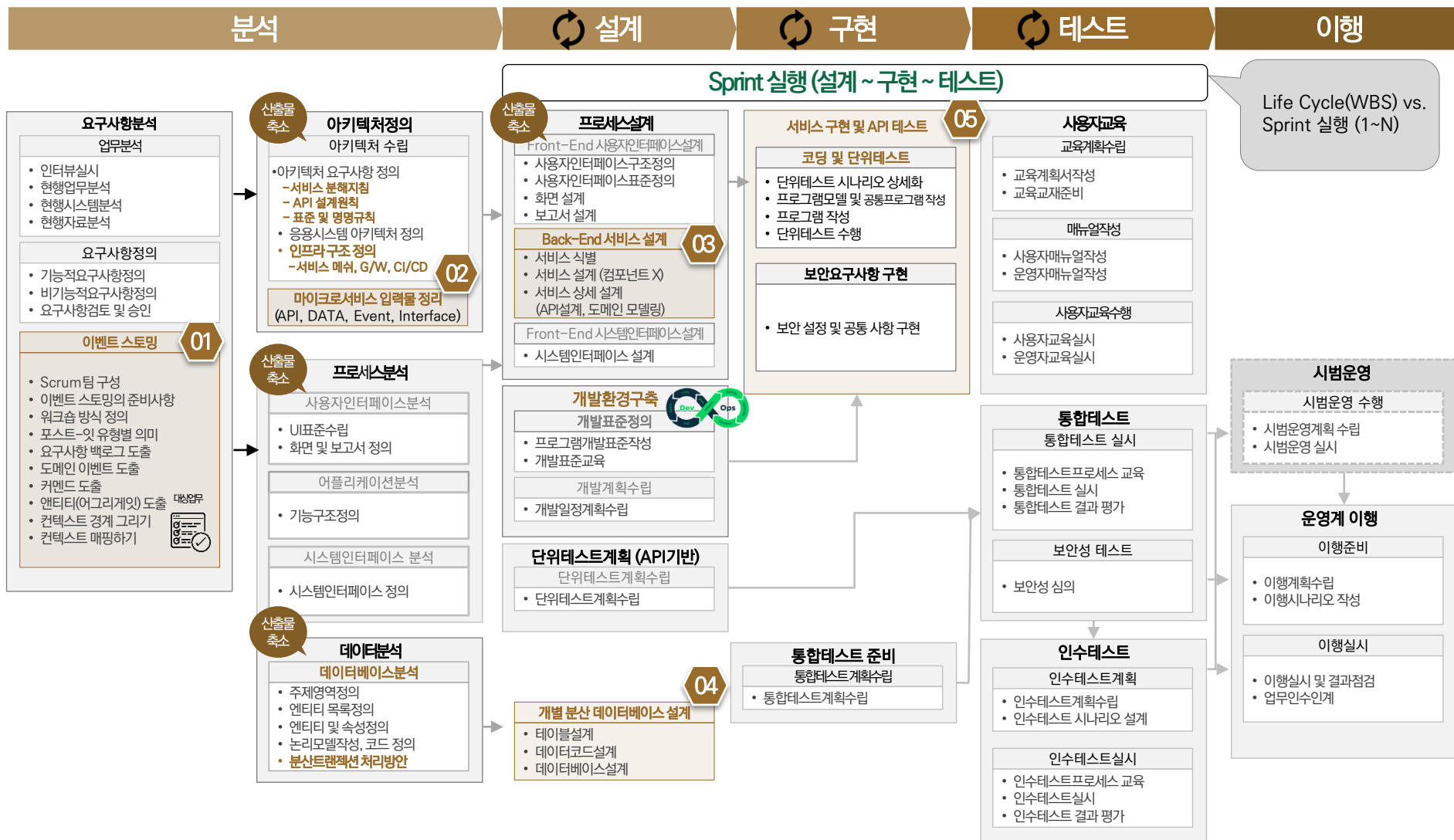
클라우드 네이티브 개발 절차 적용예시 (1/2)

CBD방법론을 기반한 마이크로 서비스 개발을 순차적으로 적용합니다.



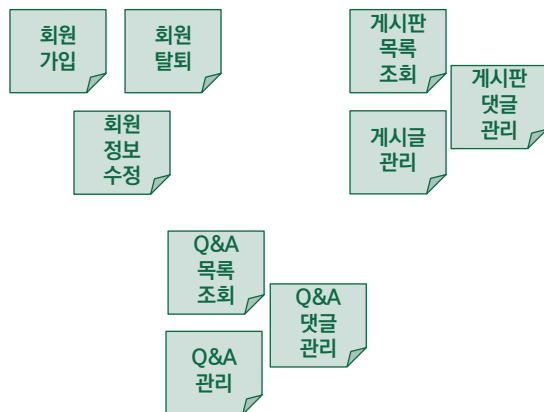
클라우드 네이티브 개발 절차 적용예시 (2/2)

애자일 방법론을 중심으로 집중 적용합니다.

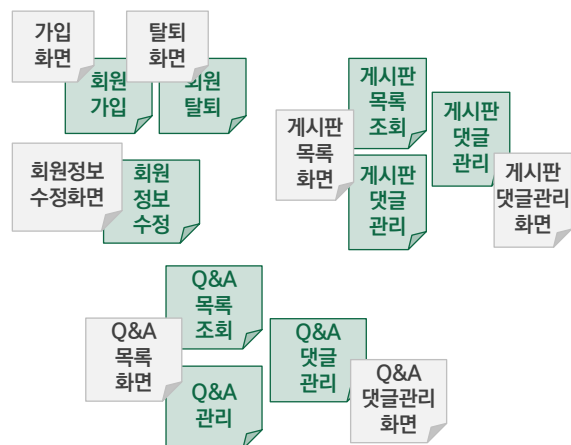


클라우드 네이티브 전환을 위한 마이크로 서비스 식별 결과 예시

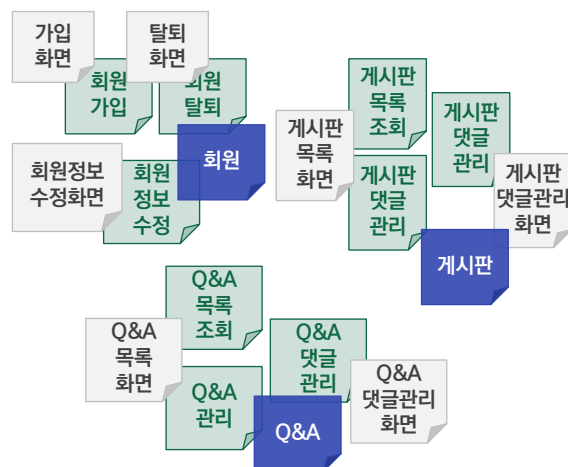
① 이벤트 도출



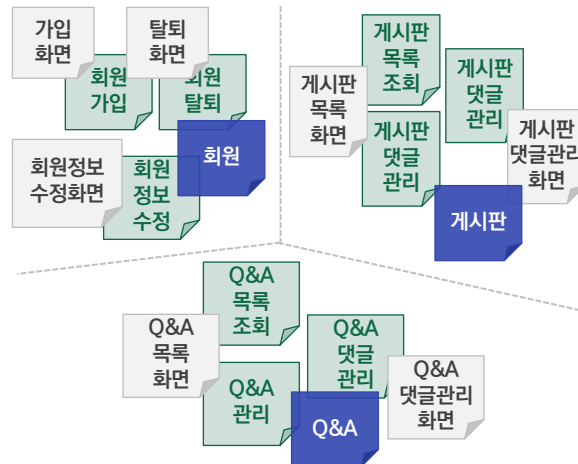
② 커멘드 도출



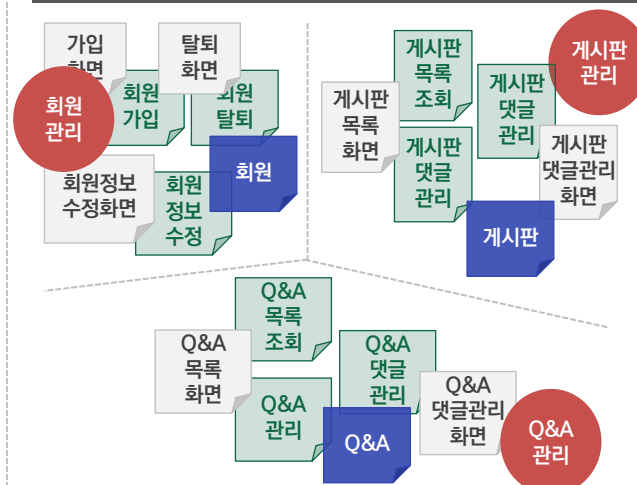
③ 엔티티(에그리게이트) 도출



④ 컨텍스트 경계 그리기



⑤ 마이크로 서비스 식별(컨테스트 매핑)



⑥ 마이크로 서비스 식별 결과



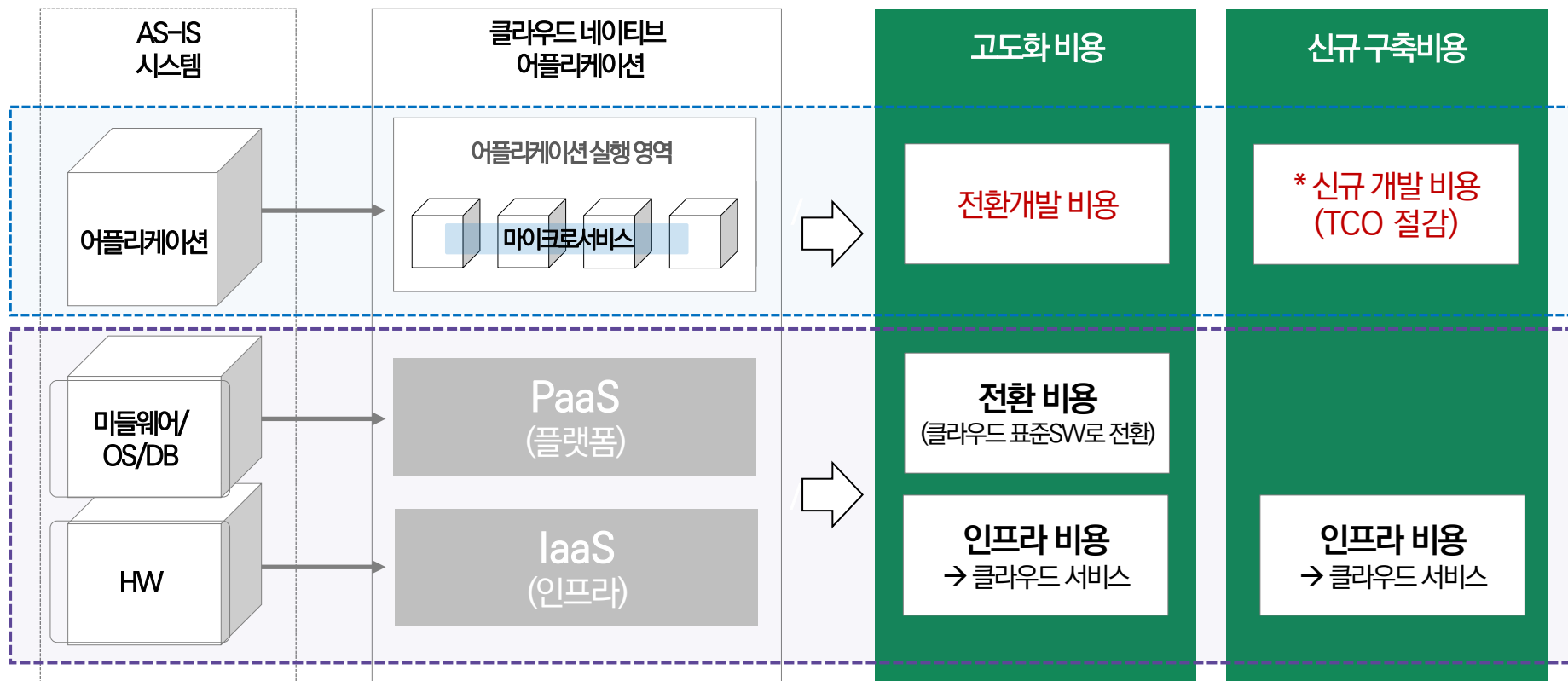
클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브 도입 비용산정은 어떻게 해야 하나요?



클라우드 네이티브 비용 산정 개요

클라우드 네이티브 어플리케이션 전환 및 신규 구축을 위한 예산을 수립한 후 발주절차를 진행합니다.



* 신규 구축 개발비용(용역비용) = 기존과 유사하게 정보시스템의 기능점수(FP)를 집계하는 방식으로 비용을 산정하되 클라우드 네이티브 어플리케이션 유형을 반영하도록 보정계수의 조정이나 기능점수(FP)의 추가

클라우드 네이티브 애플리케이션 개발 비용의 포지셔닝

‘21년 정보화전략계획 수립 공통가이드에 의하면, 클라우드 네이티브 애플리케이션 구축 시 SW 개발비 산정에 대한 검토가 필요합니다.

대분류	중분류	소분류	관련 기준	비고
시스템 구축비	HW 구매비	HW 구매/설치비	정보시스템 HW 규모산정 지침(한국정보통신기술협회)	클라우드 서비스
	SW 구축비	상용 SW 구입비		클라우드 서비스
		SW 개발비	SW사업 대가산정 가이드 (한국소프트웨어산업협회)	
	DB 구축비	DB 설계비	SW사업 대가산정 가이드(한국소프트웨어산업협회)	
		데이터 제작		
	시스템 운용환경 구축비	운용환경 설계비	엔지니어링 사업대가의 기준(산업통상자원부)	클라우드 서비스
		운용환경 공사(시설/통신망)		
	기존 시스템 이전비	HW 이전비		
		데이터 이전비		
부대비	감리비		정보시스템 감리기준(행정안전부)	

출처: 정보화전략계획(ISP) 수립 공통가이드(2021년) 참조

클라우드 네이티브 애플리케이션 구축과 관련된 개발 비용

클라우드 네이티브 애플리케이션 개발 비용 산정

클라우드 네이티브 도입 시 측정 단위는 트랜잭션과 데이터 관점에서 마이크로서비스 단위로 변화가 필요합니다.

현행 기능 분류체계

애플리케이션
(예. 총무관리 시스템)

세부 업무
(예. 회의실 관리)

단위 프로세스
(예. 회의실 예약)

마이크로서비스 기반
개발 시
업무 및 프로세스
재정의 필요

클라우드 네이티브 도입 시 기능 식별의 변화

트랜잭션(EI, EO, EQ)

FP 산정을 위한 업무 및 프로세스의 변화 예상

- 기존의 업무 및 프로세스 단위를 마이크로서비스 단위로 변경할 것인지 의사결정 필요
- 마이크로서비스 기반 개발 시 업무 및 프로세스가 세분화될 수 있음
- 기관의 결정에 따라 기존의 업무 및 프로세스 체계대로 작업할 수 있음 (단, 마이크로서비스 관련 백엔드 작업에 대한 작업 비중은 보정계수로 조정할 수 있음)

데이터(ILF, EIF)

데이터 처리 및 연계 관련 프로세스의 변화 예상

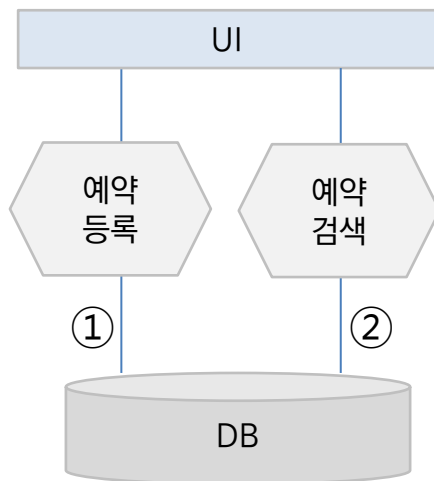
- 일반적으로 마이크로서비스 단위별로 자체 데이터를 관리하므로 내부데이터 처리(ILF)와 외부데이터 처리(EIF) 관련 기능점수가 높아질 수 있음
- 기관에서 기존의 데이터 처리방식대로 FP 작업을 수행할 가능성 있음

클라우드 네이티브 애플리케이션 개발 비용 산정

사용자 관점의 트랜잭션 처리 관련 기능점수의 변화는 없지만,
API 게이트웨이를 통한 데이터 처리 관련 기능유형의 증가가 예상됩니다.

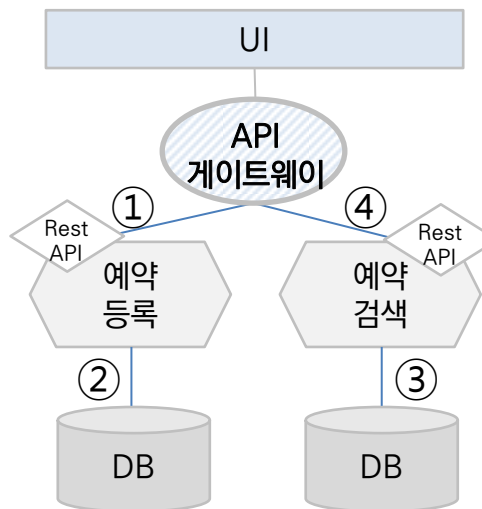
마이크로서비스 아키텍처에서의 기능 식별 방안

모놀리식 아키텍처



VS.

마이크로서비스 아키텍처 ✓



단위 프로세스	FP 유형
① 예약 등록	EI
② 예약 검색	EQ

단위 프로세스	FP 유형
① 예약 등록 API 연계	ILF
② 예약 등록	EI
③ 예약 검색	EQ
④ 예약 검색 API 연계	ILF 또는 EIF

기능 식별 시 고려사항

- 마이크로서비스 아키텍처는 API 게이트웨이를 통해 서비스 연계 작업을 수행하므로 이와 관련된 프로세스가 추가됨
 - 예약 등록 서비스의 경우, 예약등록 API 연계가 내부논리파일(ILF)로 추가됨
 - 예약 검색 서비스의 경우, 예약검색 API 연계가 내부논리파일(ILF)로 추가됨
(예약검색 DB가 없는 경우에 예약등록 DB를 읽어야 하므로 외부연계파일(EIF)로 정의함)
- 마이크로서비스 아키텍처 적용 시 모놀리식 아키텍처 대비 API 연계 프로세스의 추가에 따라 기능점수가 증가할 수 있음
→ 즉 개발비 증가가 예상됨

클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브의 아키텍처 참조모델은?

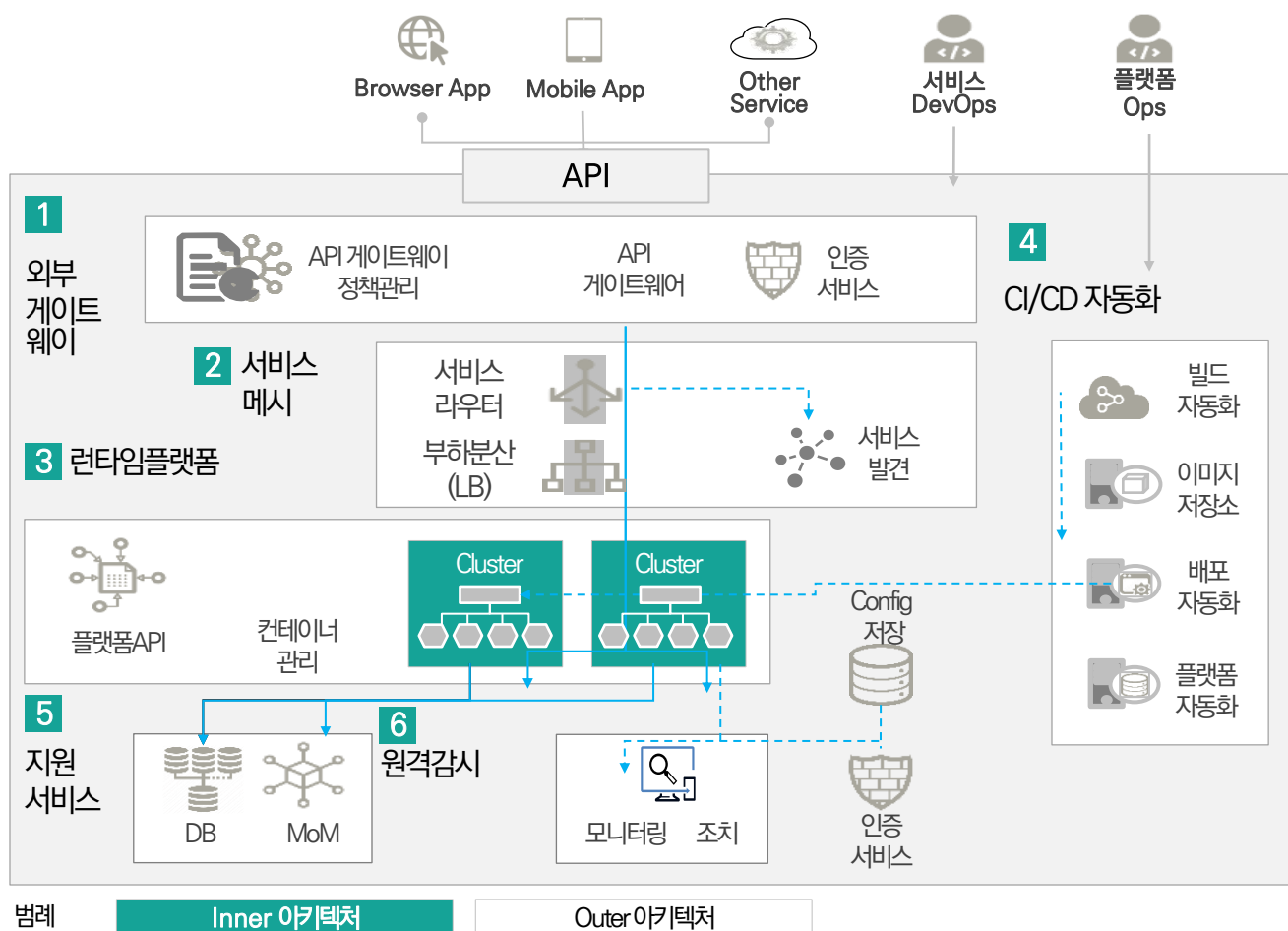


가트너 참조모델

GARTNER

가트너는 클라우드 네이티브 아키텍처 참조모델을 제공합니다.

외부 게이트웨이, 서비스 메시, 런타임 플랫폼, CI/CD, 지원, 원격감시 구성



- 1 외부로부터의 서비스 요청을 내부 구조를 드러내지 않고 처리하기 위한 구성요소
- 2 마이크로서비스 구성 요소 간의 네트워크를 제어
- 3 마이크로서비스를 실행하기 위한 컨테이너와 그 컨테이너를 관리하는 역할

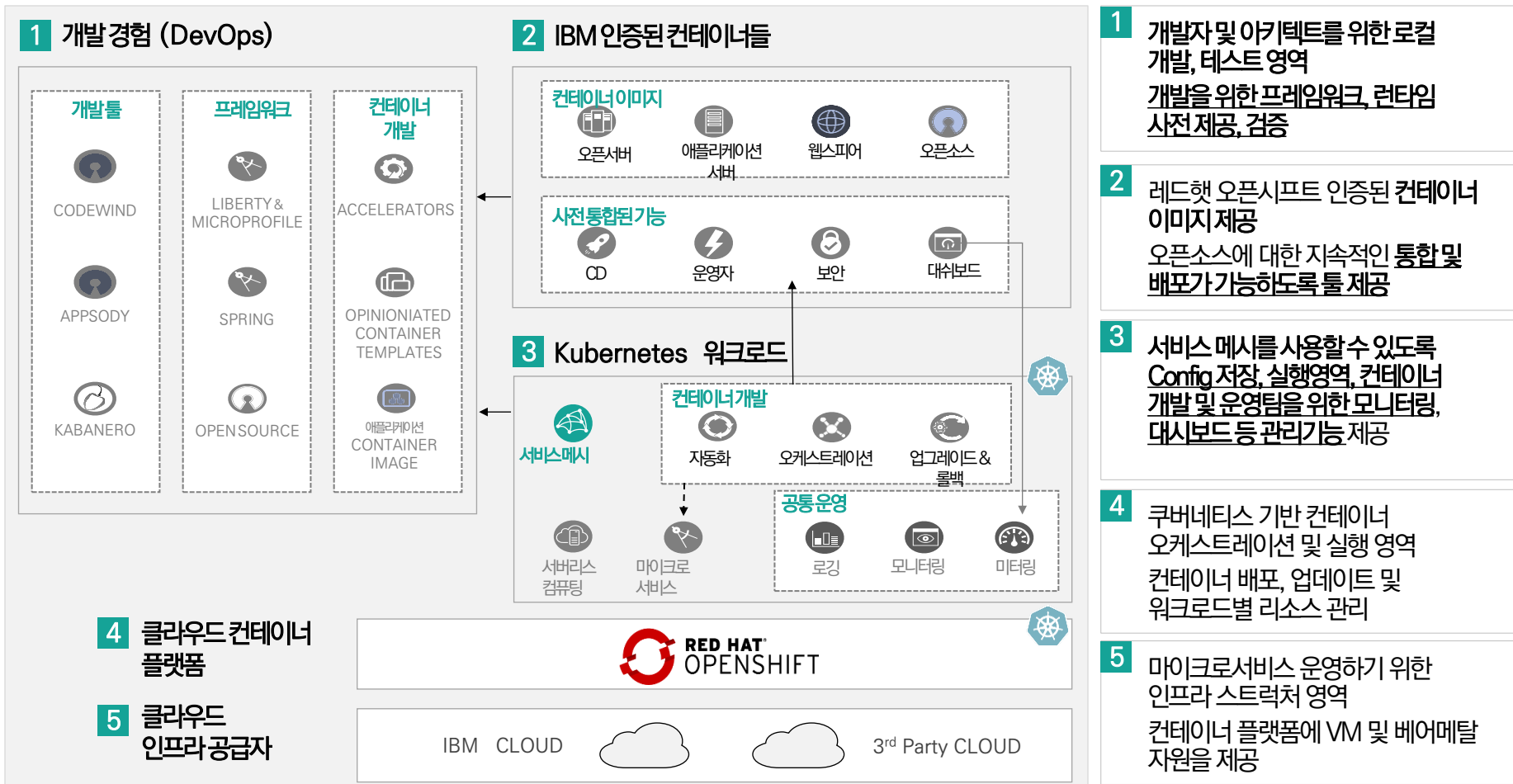
Inner 아키텍처	개별 마이크로 서비스 구축을 위한 아키텍처
Outer 아키텍처	개별 마이크로 서비스가 개발, 배포, 실행되는 전체 운영, 관리환경
- 4 마이크로서비스의 지속적인 통합, 지속적인 전달, 지속적인 배포 체계
- 5 애플리케이션이 실행하기 위해 지원되는 모든 서비스
- 6 분산환경에서 서비스 모니터링 및 헬스체크

IBM 참조모델

IBM

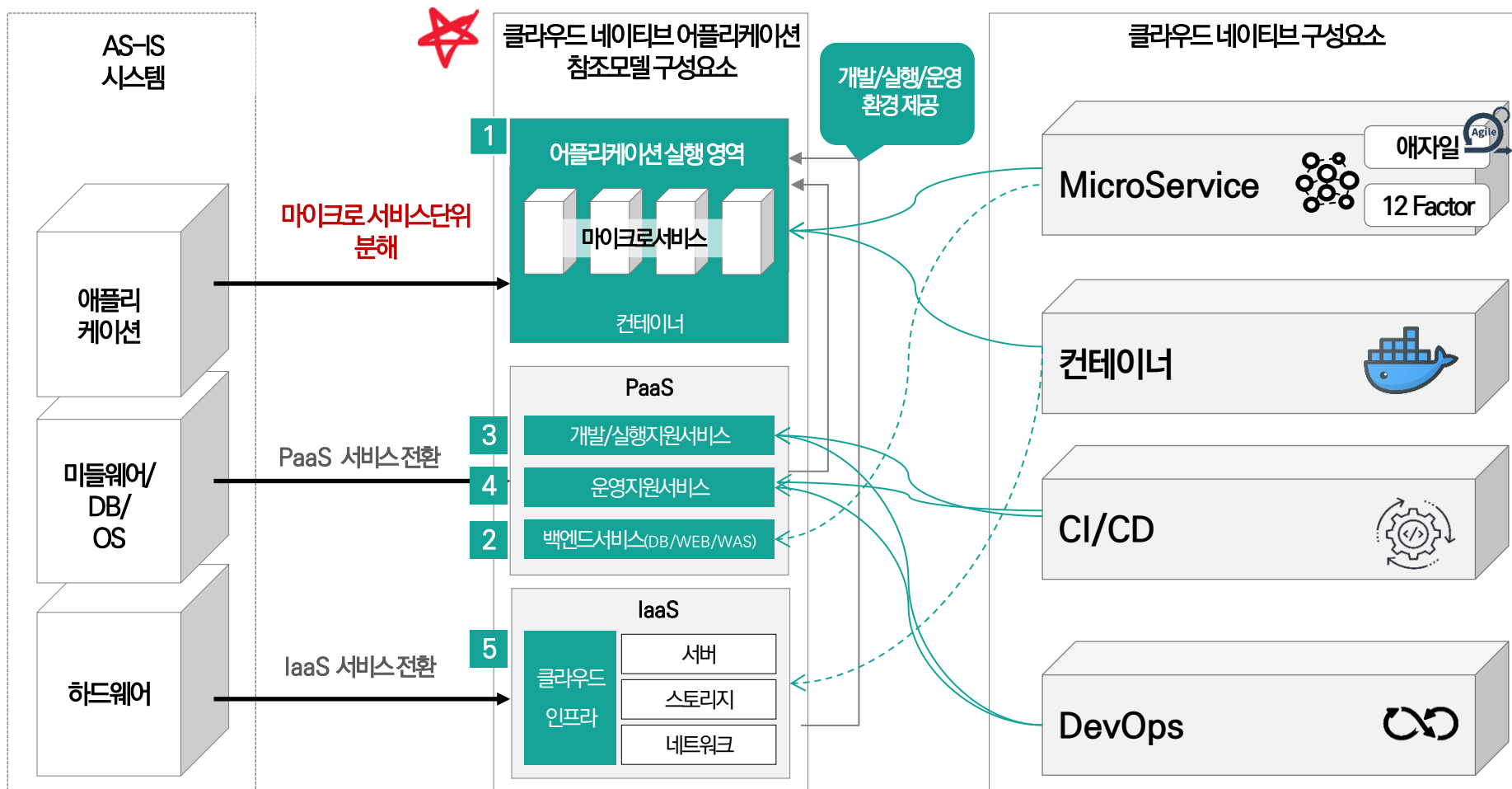
IBM은 클라우드 네이티브 아키텍처 참조모델을 제공합니다.

컨테이너/ 개발경험(DevOps) 중심으로 구성, 클라우드 인프라와 플랫폼은 서비스 임차



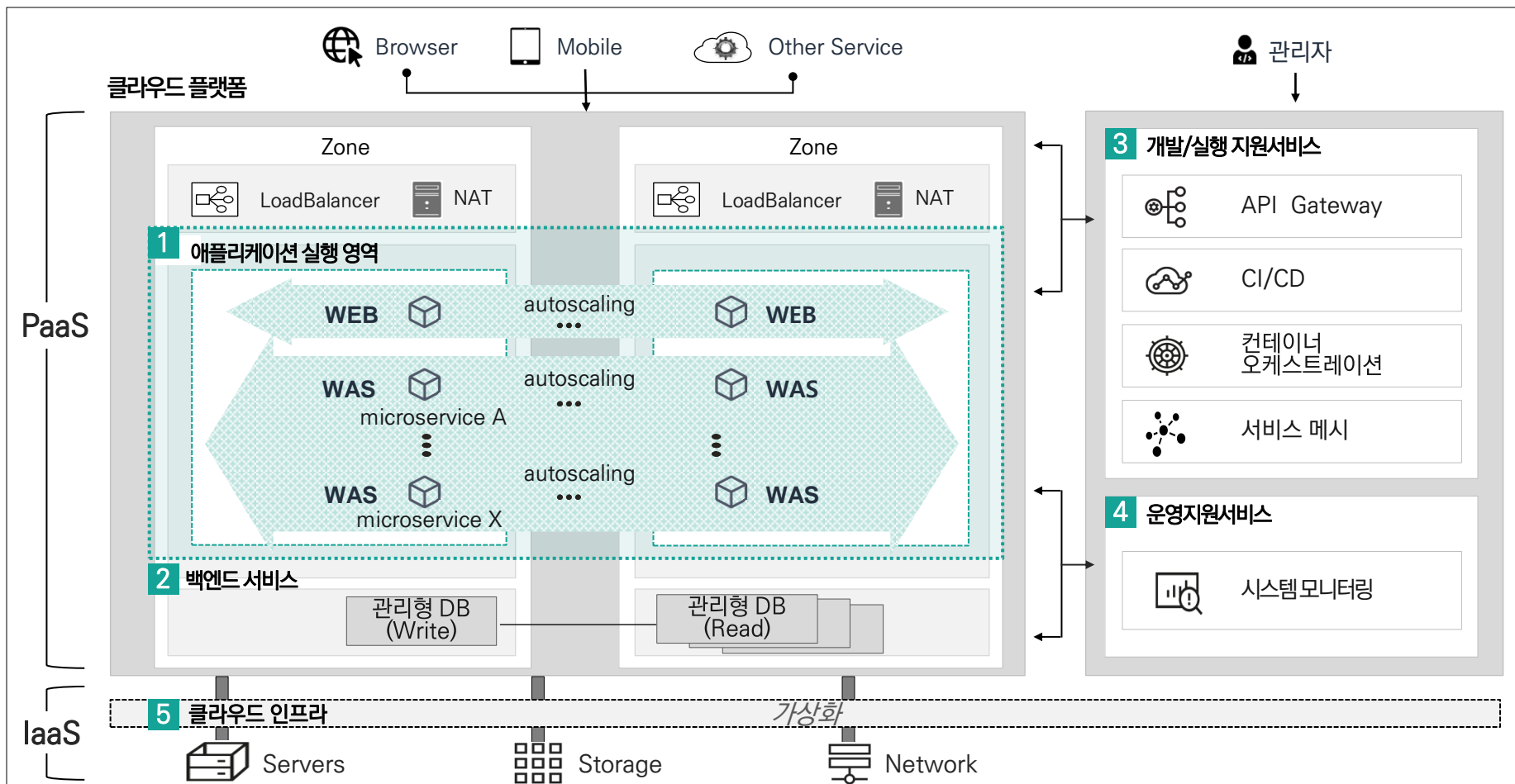
참조모델 구성요소

선진사례를 통하여 아키텍처 참조모델 구성요소를 정립합니다.



클라우드 네이티브 애플리케이션 아키텍처 참조모델

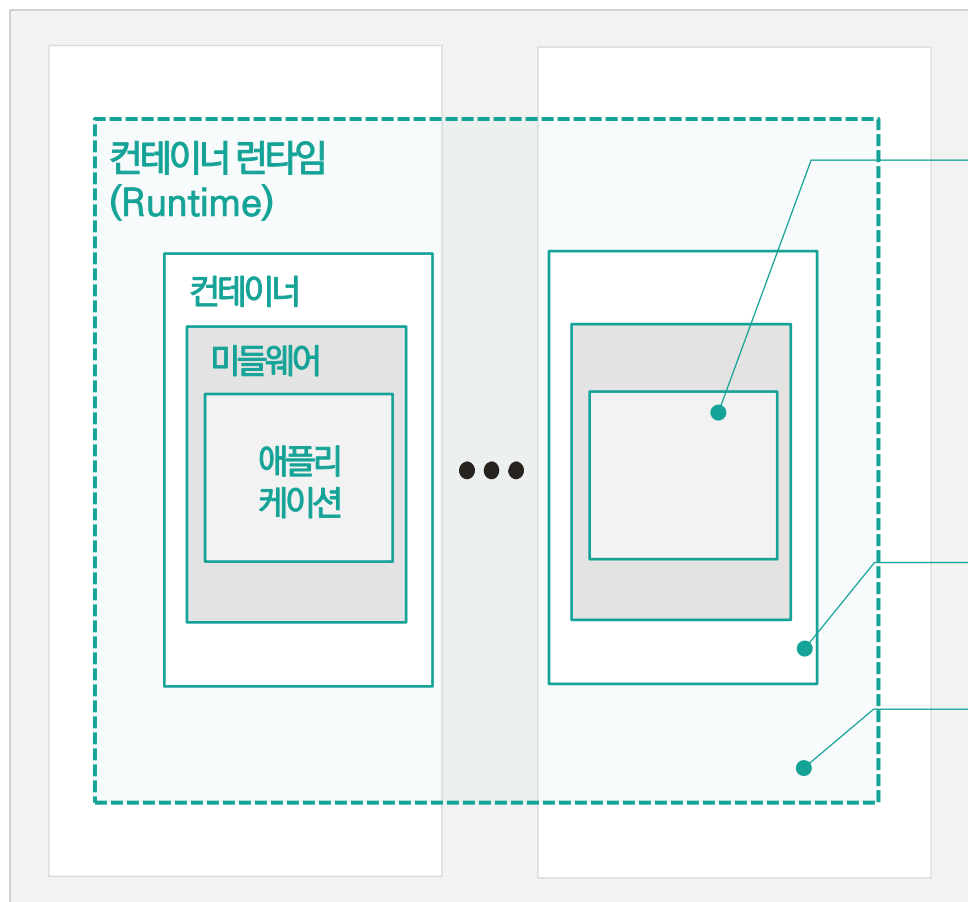
클라우드 네이티브 애플리케이션 아키텍처 참조모델 및 구성영역을 체계화합니다.



애플리케이션 실행 영역

1 아키텍처 참조모델의 애플리케이션 실행 영역입니다.

애플리케이션 실행 영역은 클라우드 네이티브 애플리케이션이 컨테이너 단위로 배치, 실행되는 영역으로 클라우드 네이티브 애플리케이션과 실행단위인 컨테이너 및 컨테이너 구동을 위한 컨테이너 런타임으로 구성됨

**애플리케이션 실행 영역 구성****클라우드 네이티브 애플리케이션**

- 미들웨어 (Web Server, WAS)를 포함하여 배포하여 개별 실행 가능
- 배포대상에 따라 아래의 2가지 컨테이너로 구성
 - Web Server : 이미지 등 정적 콘텐츠 서비스
 - WAS : 마이크로서비스 단위 애플리케이션 서비스

컨테이너 (Container)

- 클라우드 네이티브 애플리케이션 배포 및 실행 단위

컨테이너 런타임 (Container Runtime)

- 클라우드 실행환경 제공
예) 도커, 쿠버네티스 등

백엔드 서비스 영역

2 아키텍처 참조모델의 백엔드 서비스 영역입니다.

클라우드 네이티브 애플리케이션을 실행하기 위해 네트워크로 연결된 모든 리소스를 백엔드 서비스라고 하며, 클라우드 플랫폼에서 제공하는 서비스, 외부 연계 서비스 및 직접 구축한 서비스를 모두 포함함

**백엔드 서비스 종류**

DB	• 클라우드 플랫폼에서 제공 및 관리되는 DB 서비스
Cache	• 세션관리, 성능향상 등의 용도를 위한 메모리 캐싱 서비스
MoM	• 비동기 메시징(Messaging) 서비스
외부 서비스	• 인증, 간편결제, OpenAPI 형태 외부 연계 서비스
내부 서비스	• AI, 빅데이터, 챗봇 등의 클라우드 플랫폼 제공 서비스

클라우드 네이티브 애플리케이션을 수정하지 않고 다른 백엔드 서비스로 전환이 가능하도록 구성되어야 함

개발·실행 지원서비스 영역

3 아키텍처 참조모델의 개발·실행 지원서비스 영역입니다.

클라우드 네이티브 애플리케이션을 실행 환경에서 효율적으로 배포하고 안정적으로 운영하기 위해서는
클라우드 네이티브 애플리케이션에 최적화된 개발 및 실행환경을 구성해야 함

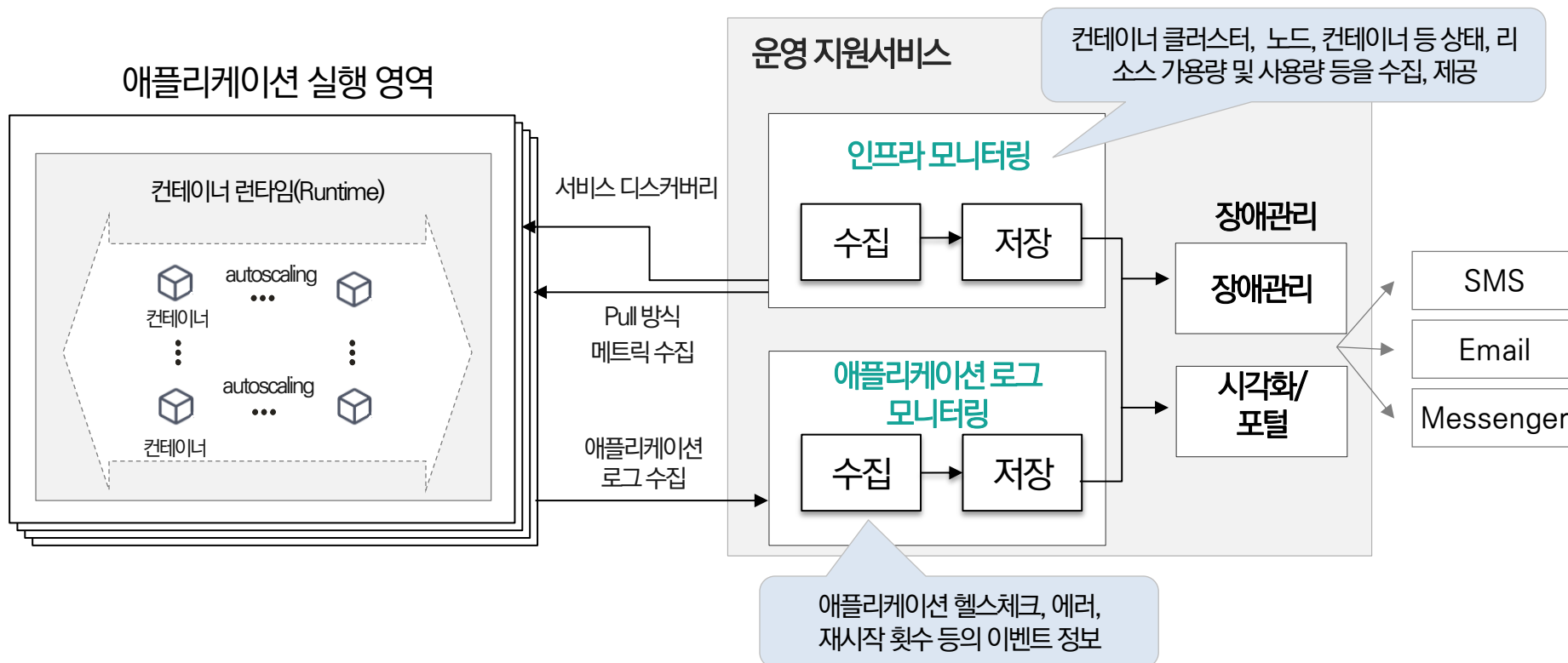
구분	지원서비스	서비스 내용	주요 솔루션
실행 환경	API Gateway	API 형태의 서비스를 제공하는 마이크로서비스 앞단에서 엔드포인트를 단일화하여 외부 사용자에게 제공 API에 대한 인증과 인가 및 여러 서버로 라우팅 하는 기능 등을 담당함	Spring Cloud Gateway, zuul, kong 등
	컨테이너 오케스트레이션	여러 서버에 걸쳐 다수의 컨테이너에 대한 배포, 컨테이너 단위의 Auto scaling, 자원할당, 장애 복구 및 모니터링 등의 운영 자동화를 제공하는 서비스	Kubernetes, Cloud Foundry, Docker Swarm 등
	서비스 메시	대규모의 마이크로서비스를 실시간으로 제어, 관리서비스 간에 통신을 하기 위해 Service discovery, 로드밸런싱, 장애 제어, Config 관리 등 제공	Spring Cloud, Istio 등
개발 환경	CI/CD	개발, 테스트, 배포 프로세스에 대한 도구 기반 자동화 및 모니터링 제공 CI(Continuous Integration)는 코드 변경 사항이 지속적으로 통합(Continuous Integration)하는 것이고 CD(Continuous Delivery)는 지속적인 서비스 제공(Continuous Delivery)을 의미	Jenkins, bamboo, git, svn 등

개발 및 실행환경 구성을 위해서 지원서비스의 활용도가 높아지고 있으며
1) 클라우드 플랫폼에서 제공되는 서비스를 이용하거나 2) 오픈소스 및 상용 SW를 활용하여 자체 구축이 가능함

운영 지원서비스 영역

4 아키텍처 참조모델의 운영 지원서비스 영역입니다.

운영 지원서비스는 클라우드 네이티브 애플리케이션 및 인프라 리소스 (CPU, Memory 등)에 대한 로그 및 사용데이터를 수집, 분석, 시각화 등 애플리케이션 전반의 모니터링 기능을 제공함



클라우드 네이티브 애플리케이션은 마이크로서비스 단위로 작아지고 OS 수준에서 가상화 된 환경에서 모니터링 대상 컨테이너 또한 동적으로 생성되므로 기존 모니터링과 다르게 서비스 디스커버리 및 Pull 방식의 데이터 수집이 적용됨

클라우드 인프라 영역

5 아키텍처 참조모델의 클라우드 인프라 영역입니다.

클라우드 네이티브 환경에서 클라우드 인프라 위에
컨테이너와 같은 경량화된 가상화 기술 및 네트워크 및 스토리지 가상화를 적용하여 클러스터를 구축함



서버, 스토리지, 네트워크 장비
등과 같은 인프라 구성요소로
클라우드의 IaaS 에 해당하며,
클라우드 네이티브 애플리케이션은
컨테이너 기반의 실행환경 외에도
마이크로서비스 구조의 애플리케이션을
개발, 운영할 수 있는
클라우드 플랫폼 서비스(PaaS)가
갖춰진 클라우드 환경이 필요함

구성요소, 방법론 및 원칙 등에 관련된 용어설명

- **MSA(Micro Service Architecture)** : 하나로 구성된 어플리케이션을 여러 개의 작고 느슨한 형태의 서비스로 쪼개어 독립적으로 서비스하고 배포할 수 있도록 구성하는 아키텍처



- **컨테이너** : 호스트 서버의 운영체제 수준의 경량화된 가상화 기술 및 그 결과물인 격리된 인스턴스(빠른 수평적 확장)



- **도커(Docker)** : 리눅스 컨테이너 기술을 적용한 가장 대표적인 컨테이너 제공 솔루션



- **쿠버네티스** : 컨테이너 오케스트레이션 도구들이 출현하였으며 대표적인 도구



- **DevOps** : 개발과 운영 프로세스의 통합 및 자동화를 통해 신속한 서비스를 제공하도록 지원하는 통합프로세스, 조직문화, 도구 등을 포함한 체계



- **DevOps 툴체인 (Tool Chain)** : DevOps를 효과적으로 적용하기 위해서 DevOps 전 프로세스를 여러 가지 도구(Tool)로 연결하여 자동화하는 과정



- **CI/CD** : 반복적인 개발, 테스트, 배포 과정에 대한 자동화와 모니터링을 제공하는 도구기반 프로세스



- **애자일(Agile)** : 사용자의 요구사항을 이해, 신속하게 반영하기 위해 기획, 설계 과정에 많은 시간과 노력을 기울이지 않고 빠르게 프로토타입을 개발하여 사용자의 피드백과 방향성을 확인하고 지속적인 개선 과정을 짧은 주기로 반복하는 개발방법론



- **12 Factor** : 클라우드 환경에 최적화되어 클라우드 모든 기능을 제대로 사용할 수 있게 위해서 지켜야 할 규칙

12 Factor

클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 온라인 설명회

클라우드 네이티브의 아키텍처 구성요소는?



컨테이너 - 컨테이너 정의 및 구조

컨테이너

컨테이너는 어떤 환경에서나 실행하기 위해 필요한 애플리케이션 코드, 런타임 모듈, 라이브러리 등의 모든 요소를 포함하는 경량화된 소프트웨어 패키지임

■ 기존 방식과 컨테이너 기술 비교



- 컨테이너 가상화는 기존의 서버 가상화 방식과 달리 컨테이너별로 실행환경을 격리하여 가볍고 빠른 서버환경을 제공하며, 하나의 물리서버에 많은 수의 컨테이너를 필요 시 즉각 생성할 수 있음
- 좀 더 효율적이고 경량화된(light weight) 애플리케이션 실행 환경에 대한 요구가 증가함에 따라 가상화 방식은 기존의 가상화 기술인 하이퍼바이저 방식에서 컨테이너 엔진 기반 가상화 방식으로 전환되고 있음

DevOps & CI/CD (1/2) – DevOps 개요

DevOps

- DevOps는 개발(Development)과 운영(Operatios)의 합성어로, 개발과 운영 프로세스의 통합 및 자동화를 통해 신속한 서비스를 제공하도록 지원하는 **통합프로세스, 도구, 조직문화** 등을 포함한 체계
- 개발팀과 운영팀의 협업이 가능해져 업무 병목 구간을 최소화하고, 애플리케이션을 신속하게 개발, 배포할 수 있음

DevOps = 개발과 운영의 협업체계



개발단계(Dev)

운영 단계(Ops)

프로
세스

계획

개발
(코딩)

빌드

테스트

릴리즈

배포

운영

모니터링

도구
(툴체인)

Jira Software

Confluence

git

Maven

Gradle

JUnit

Jenkins

docker

Jenkins

kubernetes

mantis BUG TRACKER

splunk

Nagios

조직

기획담당

개발자

테스트 담당

개발자

운영자

사용자
(요청자)

변경관리 담당

사용자
(요청자)

변경심의위원회

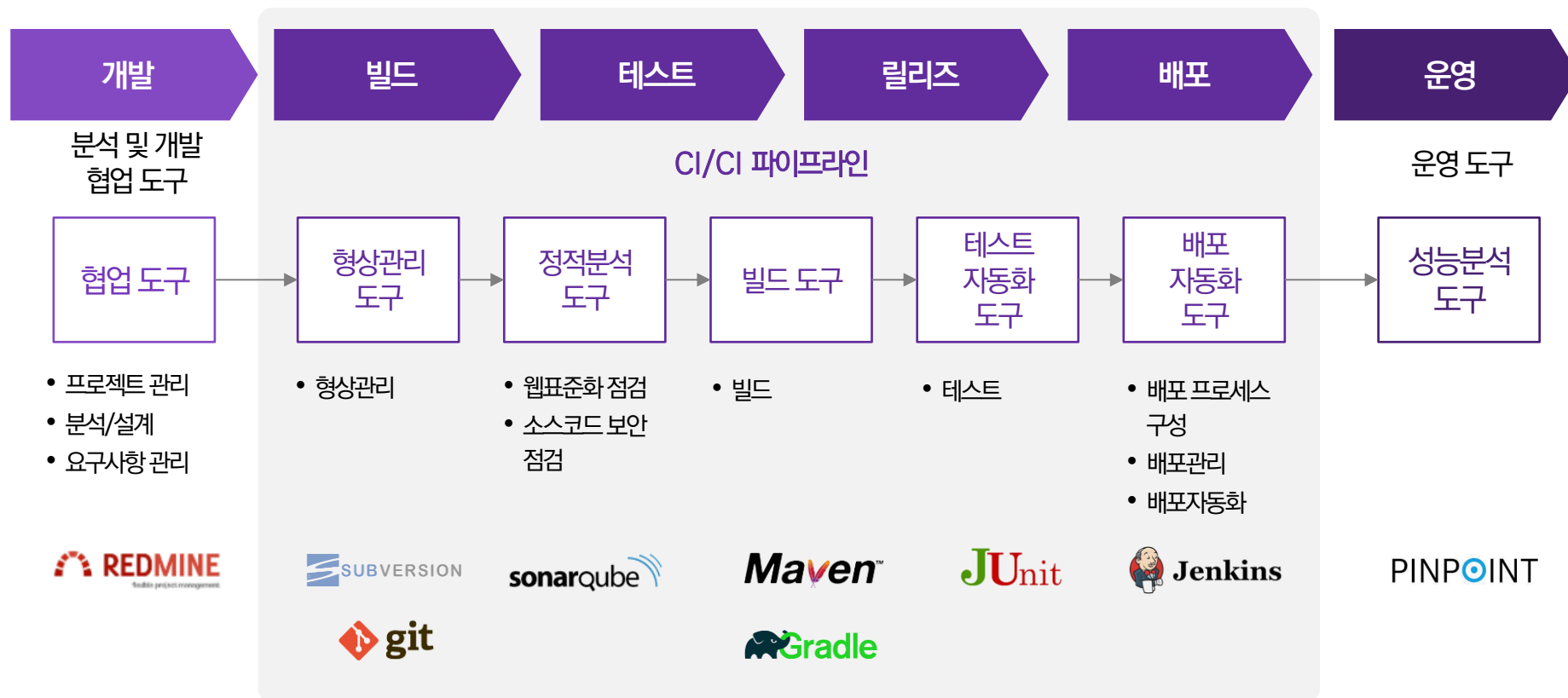
작업심의위원회

DevOps & CI/CD (2/2) – DevOps 개요

CI/CD

CI/CD 파이프라인은 개발 이후 빌드, 테스트, 릴리즈, 배포 과정에 자동화 도구를 활용하여 애플리케이션 배포주기를 단축하여, 서비스 요구에 신속하게 대응하도록 지원함

■ CI/CD 파이프라인 예시



애자일 방법론

애자일 방법론

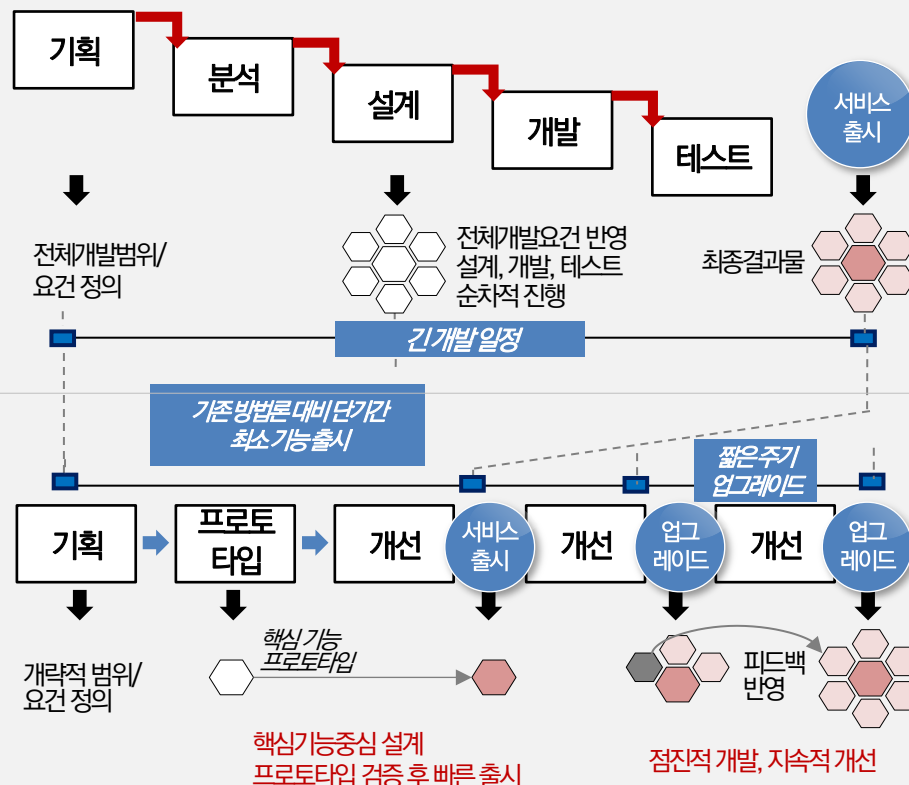
폭포수 방법론의 개발 공정은 기획, 분석, 설계, 개발, 테스트 단계가 위에서 아래로 순차적으로 진행되며, 애자일 방법론은 각 개발 공정을 명확하게 구분하지 않고 각 단계를 반복적으로 수행하면서 요구사항을 추가하거나 수정하면서 개발을 수행함

폭포수 방법론과 애자일 방법론 비교

폭포수 방법론



애자일 방법론



폭포수 방법론의 문제점

- 기획단계 많은 시간 소요
- 초기 요구사항 불명확, 잦은 변경
- 정형화된 절차, 방법론 적용으로 많은 산출물 발생

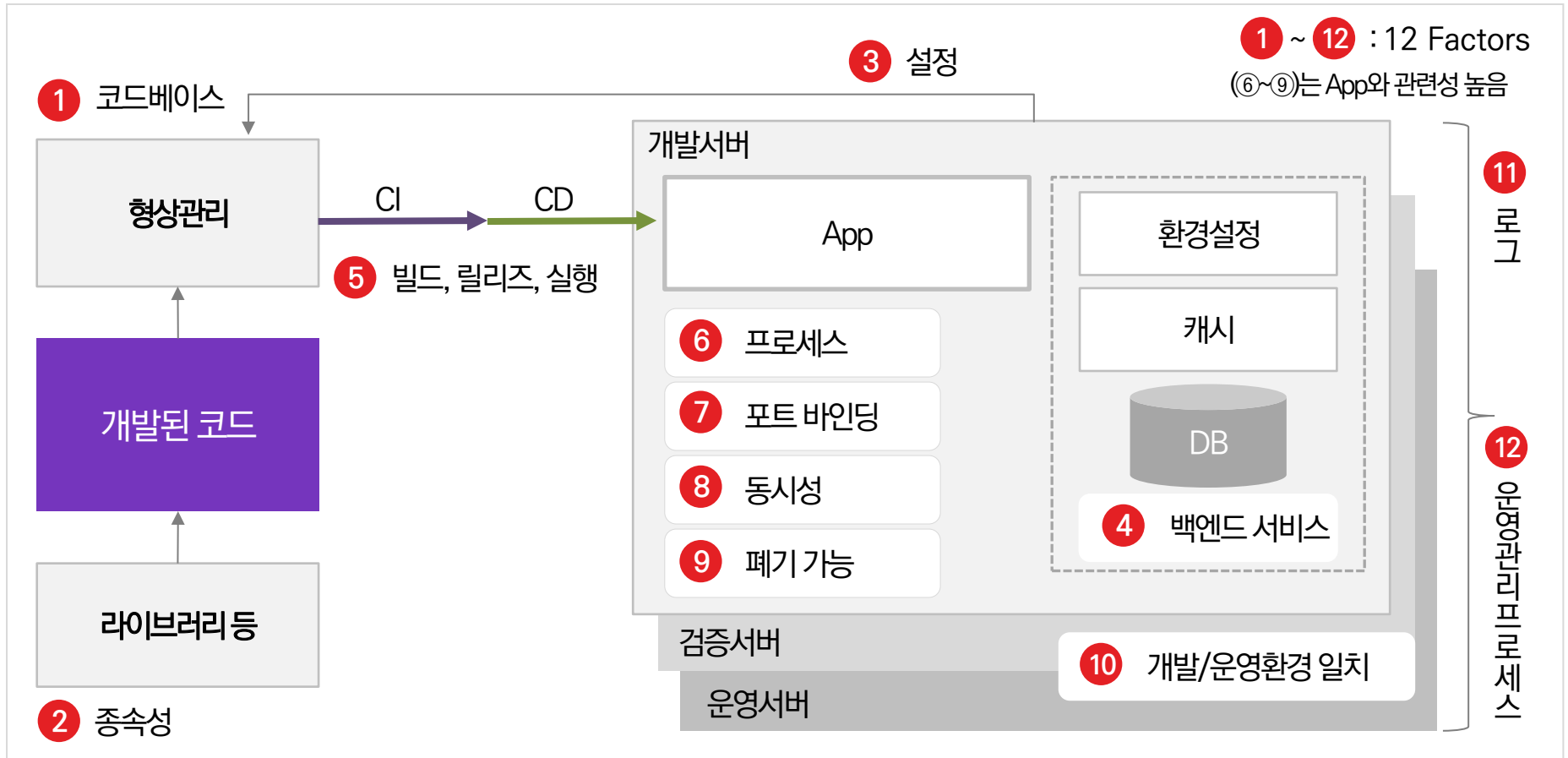
애자일 방법론 개발 방식

- 과도한 기획 지양
- 핵심기능에 집중 빠른 출시 목표
- 고객가치가 높은 기능 우선순위로 반복적으로 개선

12가지 원칙

12 Factors

12 Factors App 원칙은 2012년 Heroku에서 일하던 개발자들이 클라우드에 적합한 SaaS 애플리케이션 개발과 배포 방법에 맞는 12가지 원칙을 개념화한 것으로 클라우드 네이티브 환경에 적합하게 적용되어야 할 부분들을 명확하게 정의하고 있음



12 Factors

12 Factors

12 Factors App 원칙은 2012년 Heroku에서 일하던 개발자들이 클라우드에 적합한 SaaS 애플리케이션 개발과 배포 방법에 맞는 12가지 원칙을 개념화한 것으로 클라우드 네이티브 환경에 적합하게 적용되어야 할 부분들을 명확하게 정의하고 있음

12 Factors 원칙 설명

- | | |
|--|--|
| <p>1 코드베이스¹⁾
(Codebase)</p> <ul style="list-style-type: none"> • 하나의 코드베이스(소스코드)로 버전관리하여, 이를 여러 곳에 배포 | <p>7 포트바인딩
(Port-binding)</p> <ul style="list-style-type: none"> • 애플리케이션은 독립적이며, http 같은 포트 바인딩을 통해서 외부에 서비스 제공 |
| <p>2 종속성
(Dependencies)</p> <ul style="list-style-type: none"> • 패키지, 라이브러리 등 종속이 필요한 경우 명시적으로 선언하고 분리시켜 실행환경 종속성 제거 | <p>8 동시성
(Concurrency)</p> <ul style="list-style-type: none"> • 애플리케이션을 수평적으로 확장하며, 무상태(Stateless) 특성이 이런 확장을 단순하게 만들 |
| <p>3 설정
(Configuration)</p> <ul style="list-style-type: none"> • 소스 코드와 설정 정보를 분리, 실행시 코드에서 읽어서 사용 | <p>9 폐기가능
(Disposability)</p> <ul style="list-style-type: none"> • 빠른 시작과 그레이스풀 섯다운(Graceful shutdown)을 통한 안정성 극대화 |
| <p>4 백엔드서비스
(Backing Services)</p> <ul style="list-style-type: none"> • 애플리케이션 작동에 필요한 서비스(DB, 메시지큐, 캐시등)를 연결된 리소스로 취급하여 연결/분리가 용이 | <p>10 개발/운영 환경 일치
(Dev/Prod Parity)</p> <ul style="list-style-type: none"> • 개발계/검증계/운영계 환경을 가능한 비슷하게 유지함으로써 지속적인 개발/배포 가능 |
| <p>5 빌드, 릴리즈, 실행
(Build, Release, Run)</p> <ul style="list-style-type: none"> • 빌드, 릴리즈, 실행단계를 엄격히 분리 | <p>11 로그
(Log)</p> <ul style="list-style-type: none"> • 로그 파일을 이벤트 스트림으로 취급하여 이를 취합, 인덱싱, 분석할 수 있어야 함 |
| <p>6 프로세스
(Processes)</p> <ul style="list-style-type: none"> • 애플리케이션 실행시 하나 혹은 여러 개의 stateless 프로세스로 실행 | <p>12 운영관리 프로세스
(Admin Process)</p> <ul style="list-style-type: none"> • 시스템관리 작업은 일화성 프로세스로 만들어서 실행 |

클라우드 네이티브 한눈에 알아보기 – CNCF.io '21년

The image displays a comprehensive grid of logos for various cloud-native technologies, categorized by CNCF projects. The categories are organized into rows and columns, with each logo representing a specific project or company.

Categories and Sub-sections:

- App Definition and Development:** Database, Streaming & Messaging, Application Definition & Image Build, Continuous Integration & Delivery.
- Orchestration & Management:** Scheduling & Orchestration, Coordination & Service Discovery, Remote Procedure Call, Service Proxy, API Gateway, Service Mesh.
- Runtime:** Cloud Native Storage, Container Runtime, Cloud Native Network.
- Provisioning:** Automation & Configuration, Container Registry, Security & Compliance, Key Management.
- Special:** Kubernetes Certified Service Provider, Kubernetes Training Partner.
- Platform:** Certified Kubernetes - Distribution, Certified Kubernetes - Hosted, Certified Kubernetes - Installer, PaaS/Container Service.
- Serverless:** Serverless.
- Observability and Analysis:** Monitoring, Logging, Tracing, Chaos Engineering.

At the bottom center, the website www.cncf.io is displayed.

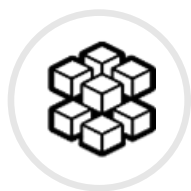
공공 클라우드 네이티브는
어떤 효과를
제공하게 되나요?



클라우드 네이티브 기대효과

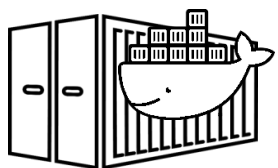
클라우드 네이티브는 빠른 배포, 컨테이너 이식성, 정책대응, 플랫폼화에 따른 표준성, 협력성, 클라우드 전환에 따른 경제성을 제공합니다.

With 클라우드 네이티브 애플리케이션



민첩성

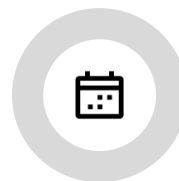
빠른 배포



이식성

컨테이너
보급

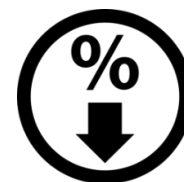
확장성

빠른 개발 및 증설
(정책대응)

표준성

플랫폼화
(소스, OS, DB등)

협력성

서로 소통하는 조직
(Agile)

경제성

서비스별 신기술 적용
(C/JAVA, DB등)

배포/테스트지연



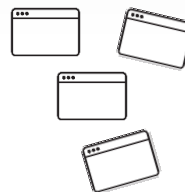
프로그램 재설치



시스템 폭주



복잡한 환경



소통 어려움



높은 비용



Without 클라우드 네이티브 애플리케이션

클라우드 네이티브 애플리케이션 행정·공공업무 변화 내역

MSA

신속한
행정

민첩성

행정·공공기관의 빈번한 제도 및 정책변화에 빠르게 대응할 수 있는
On-Demand 서비스를 제공서비스
확장성

확장성

백신예약, 마스크 예약, 연말정산, 원격근무 등 업무 폭주 시에도, 신속한 자원
증설이 가능하도록 컨테이너 기반 Scale-Out 자원환경을 제공

MSA

프로그램
이식성

이식성

기본적으로 동일한 아키텍처의 다른 컴퓨터로 전송 및 API를 호출

MSA

신서비스
보급

경제성

정부의 디지털 뉴딜(디지털 트랜스포메이션, 4차산업 혁명)에 변화에 능동적으로 대처,
다양한 언어로 서비스 개발할수 있도록 개발언어에 대한 선택성을 제공

MSA

서비스
독립성

민첩성

행정·공공기관 업무담당자의 잦은 보직 변경이 있는 상황에서, 타시스템 전체
구조를 이해할 수 없는 상태에서도 전체시스템의 영향없이 작은 서비스로 개발서비스
자동화

협력성

DevOps기반 자동화를 통한 휴먼장애 방지, 각 서비스별 독립적인 개발·배포를
통하여 행정·공공기관 운영자에게 편리한 배포 및 모니터링 환경을 제공

MSA

플랫폼
보급확산

표준성

디지털 정부 플랫폼 기반의 손쉬운 개발 툴 등 표준화된 개발플랫폼 환경제공을 통하여
공공 클라우드 센터의 전면전환을 효과적으로 수행

행정·공공 업무가 이렇게 바뀝니다.

클라우드 네이티브 기반
'포스트 코로나 시대의 디지털 정부혁신 발전계획'과
'한국판 뉴딜 종합계획'에 따른

빠른 대국민 서비스가 가능합니다

클라우드 네이티브 기반
마이크로 서비스 품질 강화(즉시 배포 등)에 따른

24 x 365 무중단 행정 서비스가 가능합니다.

클라우드 네이티브 기반
공공 클라우드 센터 전면 전환에 따른
센터, 부서, 업무 간의

상호 운용성 제공이 가능합니다.

클라우드 네이티브 기반
개발부터 운영까지 전주기 관리를 통한 (DevOps, CI/CD)

서비스 운영 자동화가 가능합니다.



클라우드 네이티브 기반 행정·공공 서비스 확산 지원
클라우드 네이티브 발주자 가이드

감사합니다.

