

기업의 Modernization 을 위한 Readiness 확보

홍장헌 팀장, 황재문 팀장

SK(주) C&C

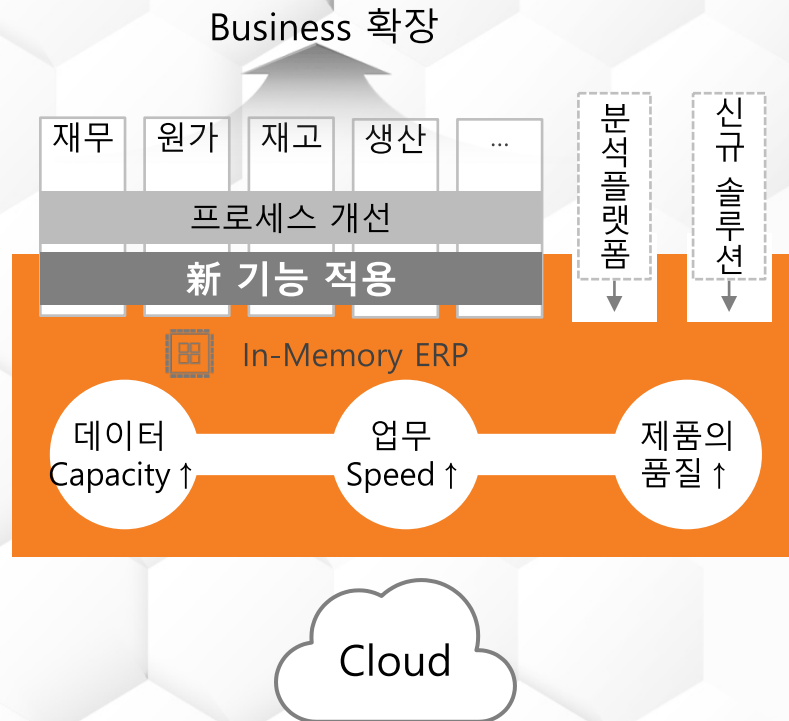
기업의 Modernization 을 위한 Readiness 확보

Part I. Cloud Modernization

황재문 팀장

SK(주) C&C

What is Modernization



- ✓ 기업이 가지고 있는 IT 기술요소를 진화하는 신기술 (Cloud, Mobility, BigData, AI) 기반으로 전환하여 현대화 하는 과정
- ✓ 비용절감, 생산성 향상, 고객 경험의 향상 추진
- ✓ Monolithic 및 On-Premise Application 중심
※) 변화에 신속한 대처가 불가능한 구조
→ Cloud Native Application 및 Architecture

Why modernize Legacy System

Legacy System



High-Cost Maintenance

신기술에 비해 복잡성 증가, 유지 관리 어려움
※) Unix 장비, System Language

Lake of Flexibility and Adaptability

신기술 접목이 용이하지 않음
※) Public 전환 제약, New Tech (Container)

Non-Agilty

새로운 Business 요구를 적응하기 불가능함
※) Silo Architecture

Challenges

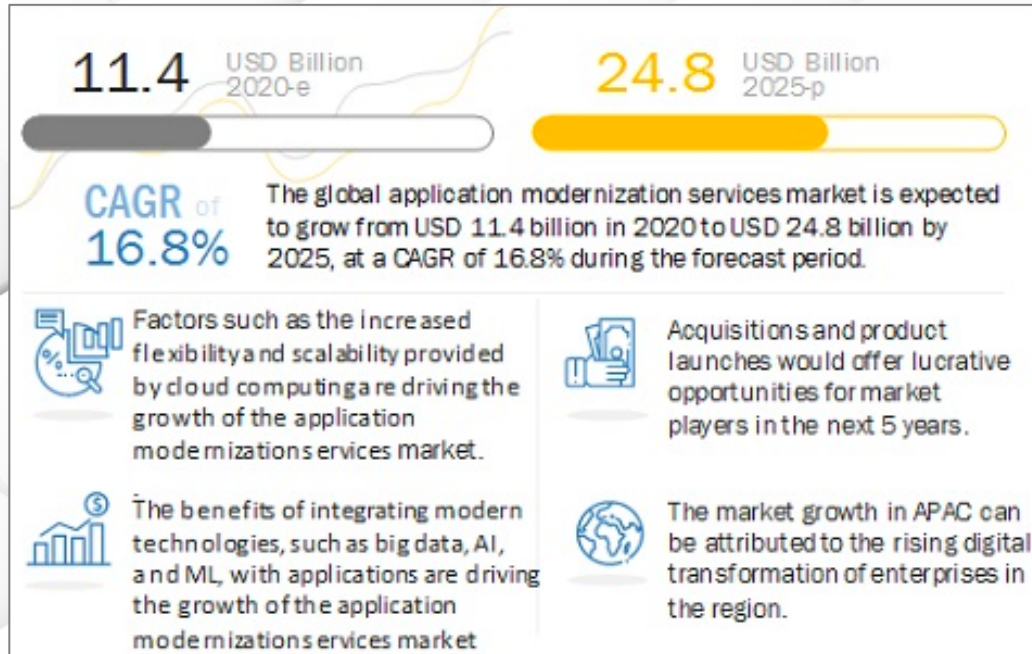
“전환 복잡성” **Risk**

“현대화의 효과” **Not Clear**

“변화를 위한” **Strong Case**

Modernization 규모

Modernization Market



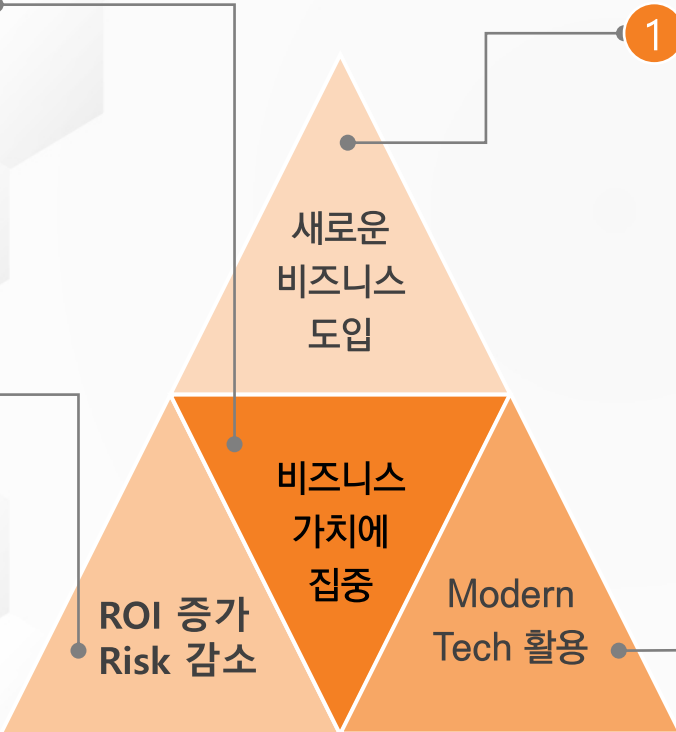
- ✓ '20 \$114억 → '25 \$248억
 - 예측 기간 동안 16.8%의 연평균 성장률
- ✓ Modernization 가속화 이유
 - Cloud 기반의 유연성/확장성 활용
 - Modern Tech 활용 (Big Data, AI, ML/DL)
 - Legacy Transforming
- ✓ Impact of COVID19
 - Remote 근무/Data 접근을 통한 생산성 저하 최소화

Modernization Values



- 비즈니스 가치에 집중** ④
- 개발 환경, 자동화된 운영 체계
 - 확장성과 장애 복구성

- ROI 증가, Risk 감소** ③
- Software Automatic Upgrade
 - Delivery 신속성
 - Legacy System의 Security 향상



- ① 현대화는 새로운 비즈니스 기능 도입 촉진**
- 미래 시장, 비즈니스 변화
 - 기술 변화/경쟁, 장기적 생존

- ② 현대화(Modern)된 Infrastructure 전환의 원동력**
- Cloud Based Infrastructure
 - Cost Reduction
 - BigData, AI 기술의 활용

Assessment

Consider
01

Legacy Application /System

기존 Enterprise 가 보유한 Legacy System 이 제공하는 가치?

Consider
02

Operating Costs

운영하는데 소요되는 비용 (인프라, 라이선스, 인력규모 등)?

Consider
03

Future performance/ Scalability

기업이 지향하는 미래의 모습에 합당한 성능과 확장 가능?

Consider
04

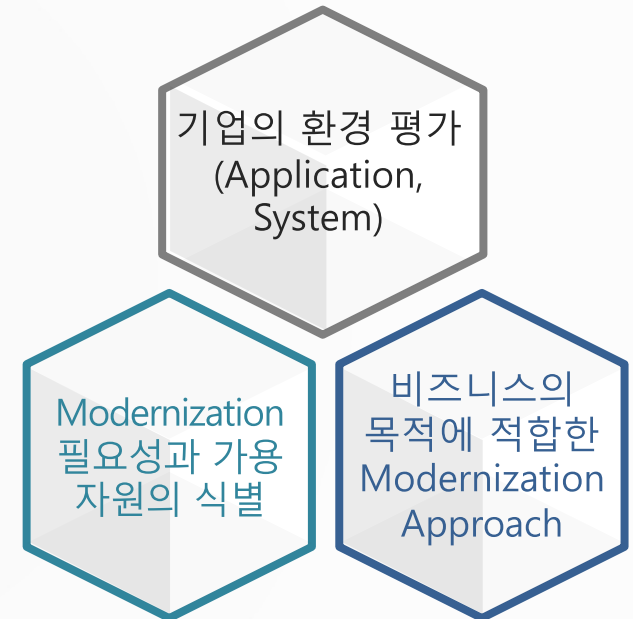
In-House Skill

향후 5년간 Legacy 시스템을 운영할 기술 역량의 보유?

Consider
05

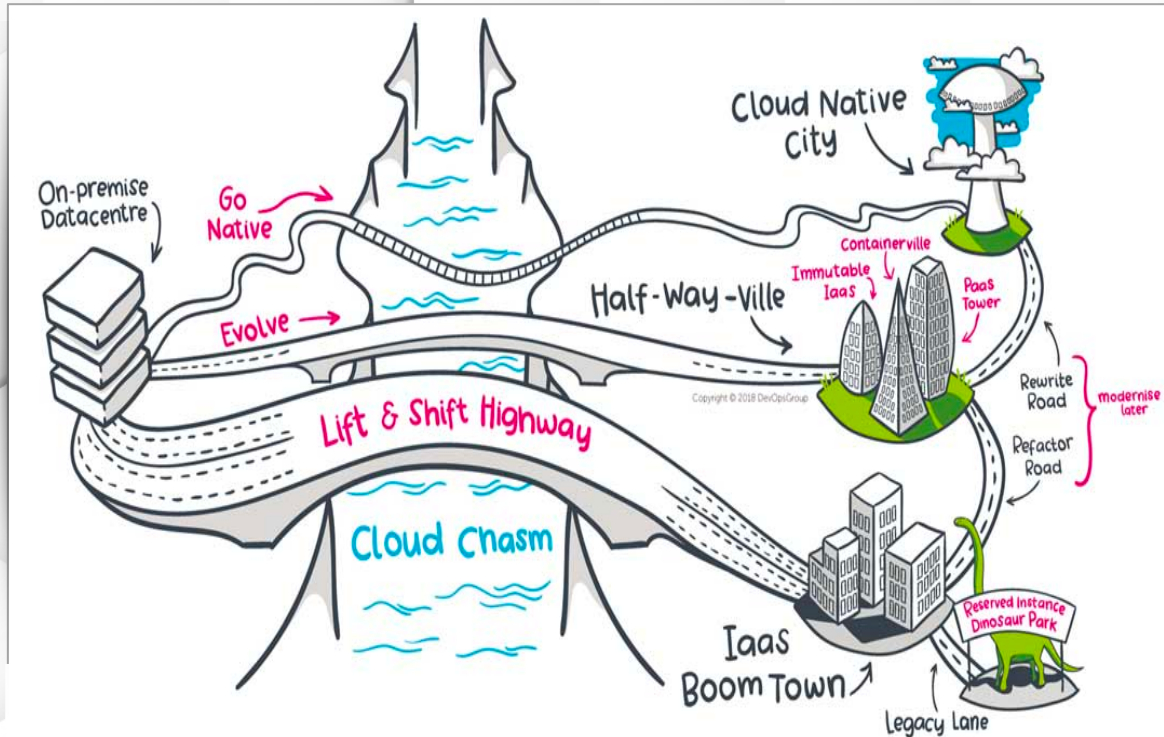
Changeability

기업의 미래 환경과 비즈니스 변화 목적에 맞는 변화 가능성?



Approaches

For Enterprise IT Modernization



Cloud Native (Rearchitecting)

Cloud Native 기술에 적합 한 구조로 재설계 및 재 개발을 추진

※) Microservice Architecture

Evolve (Replatforming)

System의 외부 기능적 변화는 없지만 기존의 코드와 아키텍처의 Optimize, Resturcturing

※) Container Platform

Lift & Shift (Rehosting)

Legacy System의 기능, 코드의 변화 없이 가상화 기술 또는 Cloud 인프라에서 재구동

Modern Technology

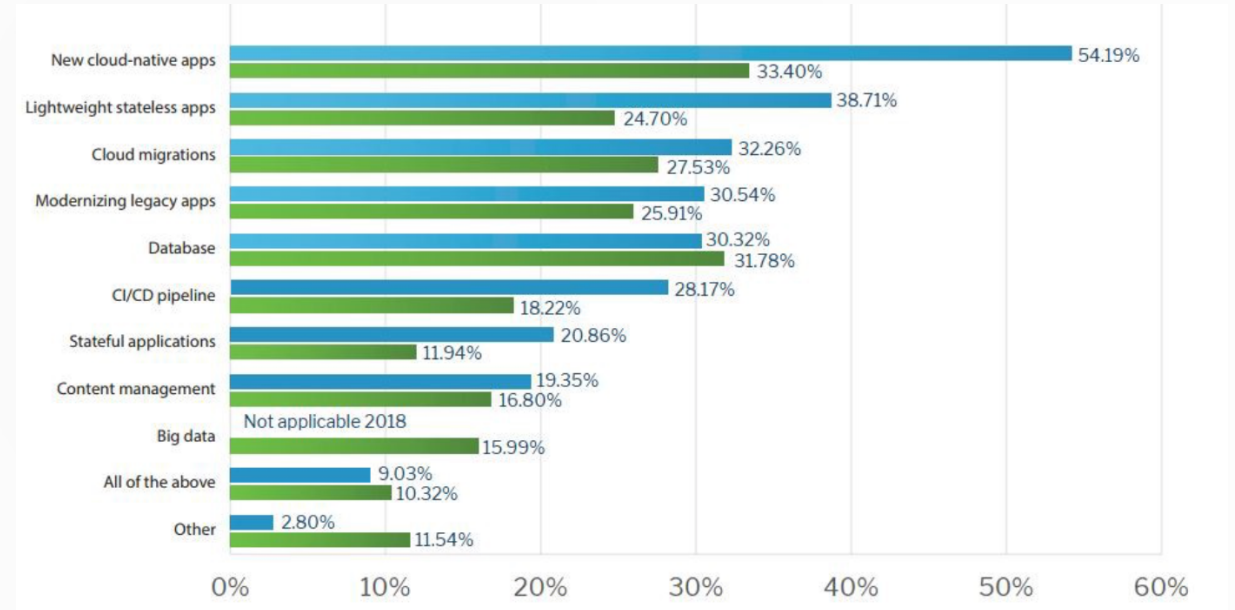
For Enterprise IT Modernization

Microservice Architecture



- 자원의 Utilization 및 활용도 증가
- 전체 장애 방지
- 서비스 단위별 개선, 축소/확대 유연성
- 변화가 빠른 시대에 대응하는 구조

Container



Cloud Native Apps, Cloud Migration, Modernizing Legacy Apps, Database, CI/CD Pipeline, Big Data

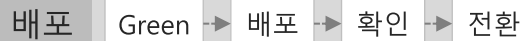
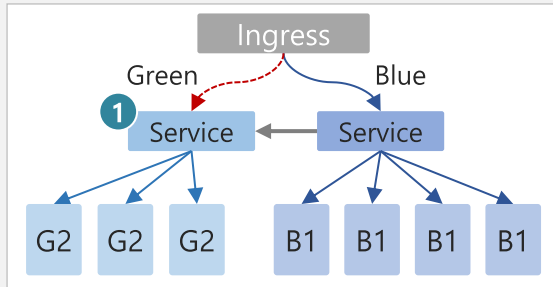
SKC&C Case

MSA, Container 기반 Modernization 사례

A社 (통신)

Mobile 상품의 판매

생산장비 증가에 따른 신속한 처리와 확장성을 극복과 효율적 운영을 위한 플랫폼 도입

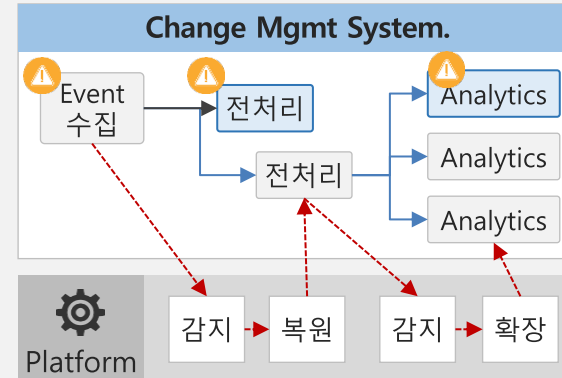


- Enterprise 운영 정책에 최적화된 배포 적용
 - 매주 목요일 배포에서 주 2회 및 단계적 축소
 - Container 기반의 신속한 배포 적용
- Container 기반의 빠른 복구 체계
 - 기존 배포 Version의 Rolling Back 처리
 - 검증 환경 전환시에 운영 상태로 확장 처리

B社 (생산)

장비 상태변화 관리

생산장비 증가에 따른 신속한 처리와 확장성을 극복과 효율적 운영을 위한 플랫폼 도입

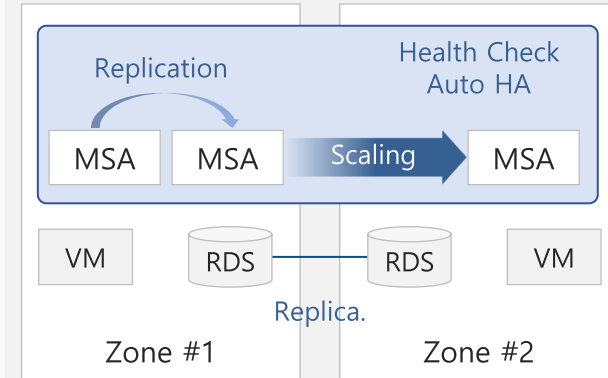


- 생산량 증가를 위한 신속한 자원 확장
 - 대규모 Event 및 Data 처리를 위해 VM 比 Container의 신속한 처리 아키텍처 도입
- 자동화된 컨테이너 자원의 운영
 - Container 상태에 따른 자동화된 Life-Cycle
 - 처리 증가에 따른 유연한 AutoScaling 적용

C社 (물류)

수출입통관, 내륙운송 및 창고관리

시스템 장애時 항공 통한 해외 운송 불가 등 장애에 고객사 막대한 영향 발생



- 시스템 장애에 빠른 대응 체계 마련
 - Container 기반의 Resilience 확보 (Auto Healing 및 신속한 복구)
 - Workload 기반의 확장 체계
- 서비스 가용성 증가 아키텍처 마련
 - Multi-Zone 기반의 전체 서비스의 가용성 적용을 통한 단일 Zone 장애 대응 체계

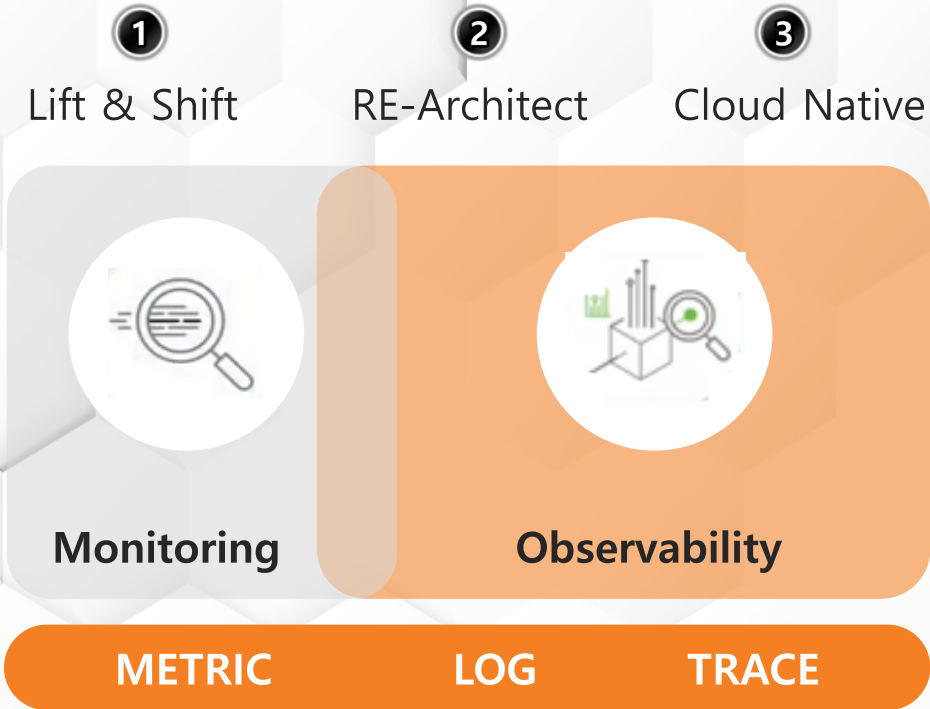
기업의 Modernization 을 위한 Readiness 확보

Part II. Cloud Service Observability

홍장헌 팀장

SK(주) C&C

What is Observability



- ✓ 전통적인 모니터링 시스템은 임계값과 변화량에 기반한 경고(Alerting)을 통해 상대적으로 고립된 환경에서 신호들을 수집하고 검사한다.
- ✓ 반면에 Observability는 개발자와 운영자 모두에게 예측하기 힘든 시스템을 더 효율적으로 설명하는데 도움을 줄 수 있다.

- GARTNER

Approaches to modern application development

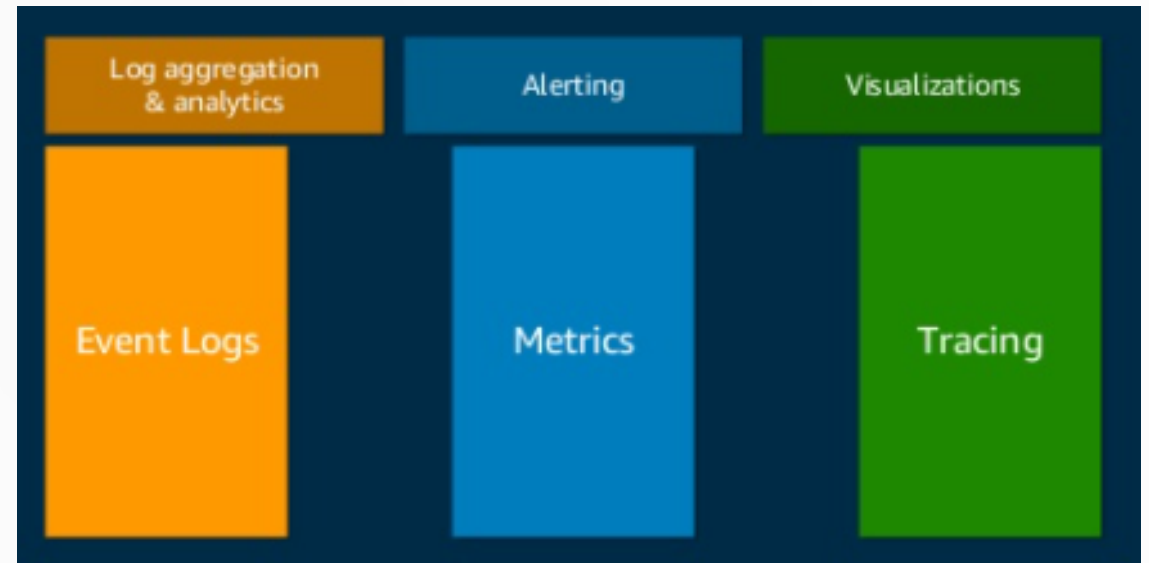
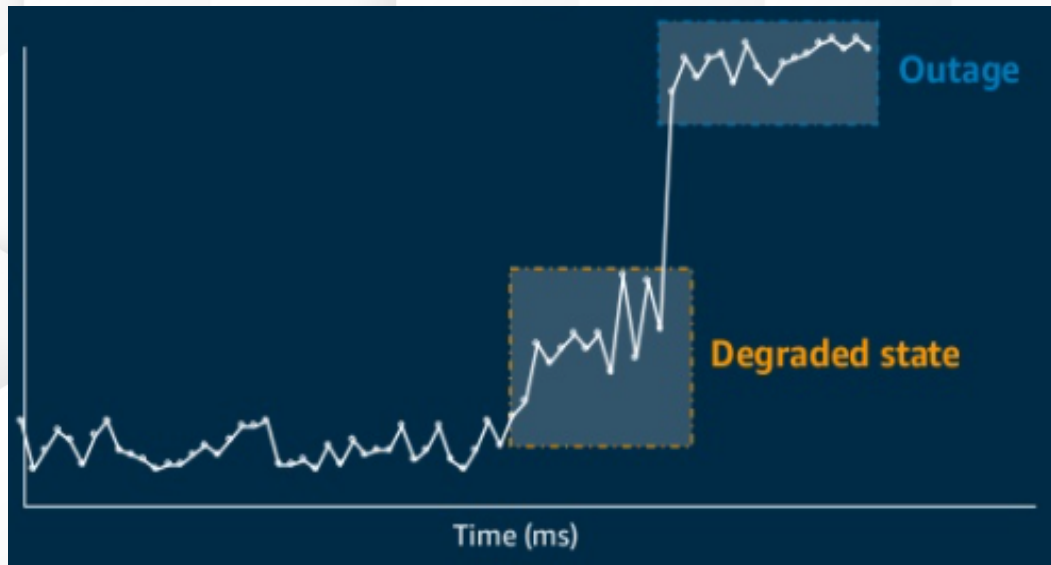
- Simplify environment management with **serverless technologies**
- Reduce the impact of code changes with **microservice architectures**
- Automate operation by **modeling applications & infrastructure as code**
- Accelerate the delivery of new, high-quality services with **CI/CD**
- Gain insight across resources and applications by enabling **observability**
- Protect customers and the business with **end-to-end security & compliance**

Observability vs Monitoring



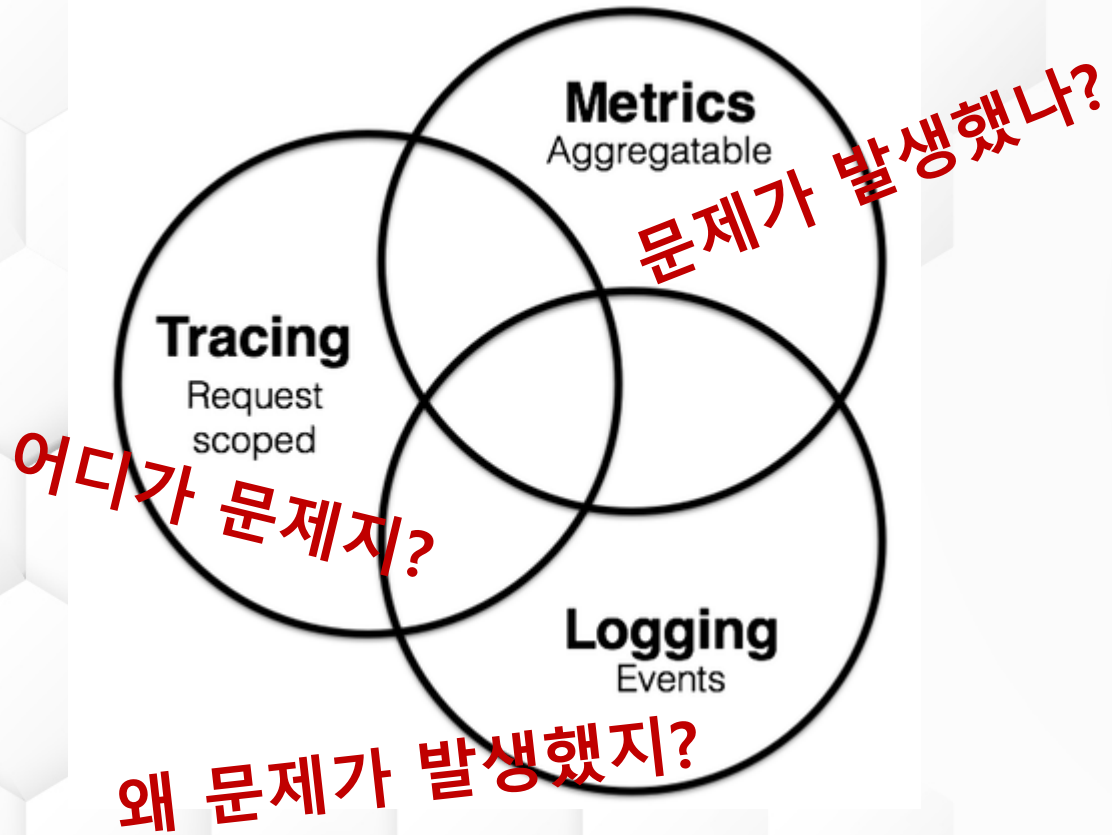
Using Observability

Proactive Operations helps Cloud service issues



Source : Distributed Systems Observability *by Cindy Sridharan*

Observability 3가지 구성요소



Metrics for Detecting

시간간격에 따라 측정된 데이터의 숫자 표현 저장, 처리, 압축 및 검색에 최적화되어 있으며, 과거의 추세를 반영하는 대시보드를 구축하는데 적합

Tracing for Troubleshooting

End-To-End 요청 흐름에 대한 일련의 사건의 표현 데이터 구조는 로그와 비슷하나 요청경로와 관련된 다양한 서비스를 이해하고 분기점과 영향을 이해하는데 도움

Logging for Root Cause

시간이 지남에 따라 일어난 개별 사건의 타임 스탬프 기록 충분한 컨텍스트와 함께 분산 시스템의 구성요소에서 나타나는 긴급하고 예측할 수 없는 동작을 발견하는데 유용

Metrics

□ 수집 대상의 다양화

- AWS, Azure, GCP, IBM 등 다양한 CSP
- 다양한 관리형 서비스 (RDB, Serverless ...)
- 혼재된 Infra환경 (Private/Public/Hybrid)

VM
Managed Svc



OnPrem

Metrics
Collector

Metrics
query engine



Dashboard

Alerts



Logging



2020-12-10T 16:38:23+00:00 ERROR Read timed out

언제 발생 했는가?

timestamp	2018-02-20T 16:38:23+00:00
-----------	----------------------------

어떤 메시지 인가?

level	ERROR	log	Event
-------	-------	-----	-------

어떤 서비스에서?

service	Registration-service	team	events
---------	----------------------	------	--------

무슨 일이 있었지?

commit	542a8b8e	build	542a8a8e.7	runtime	Java-1.8.0_161
--------	----------	-------	------------	---------	----------------

어디에서?

region	europe-west2	node	Node_e79f3e52
--------	--------------	------	---------------

누가 원인인가?

CustomerID	55123	UserID	458
------------	-------	--------	-----

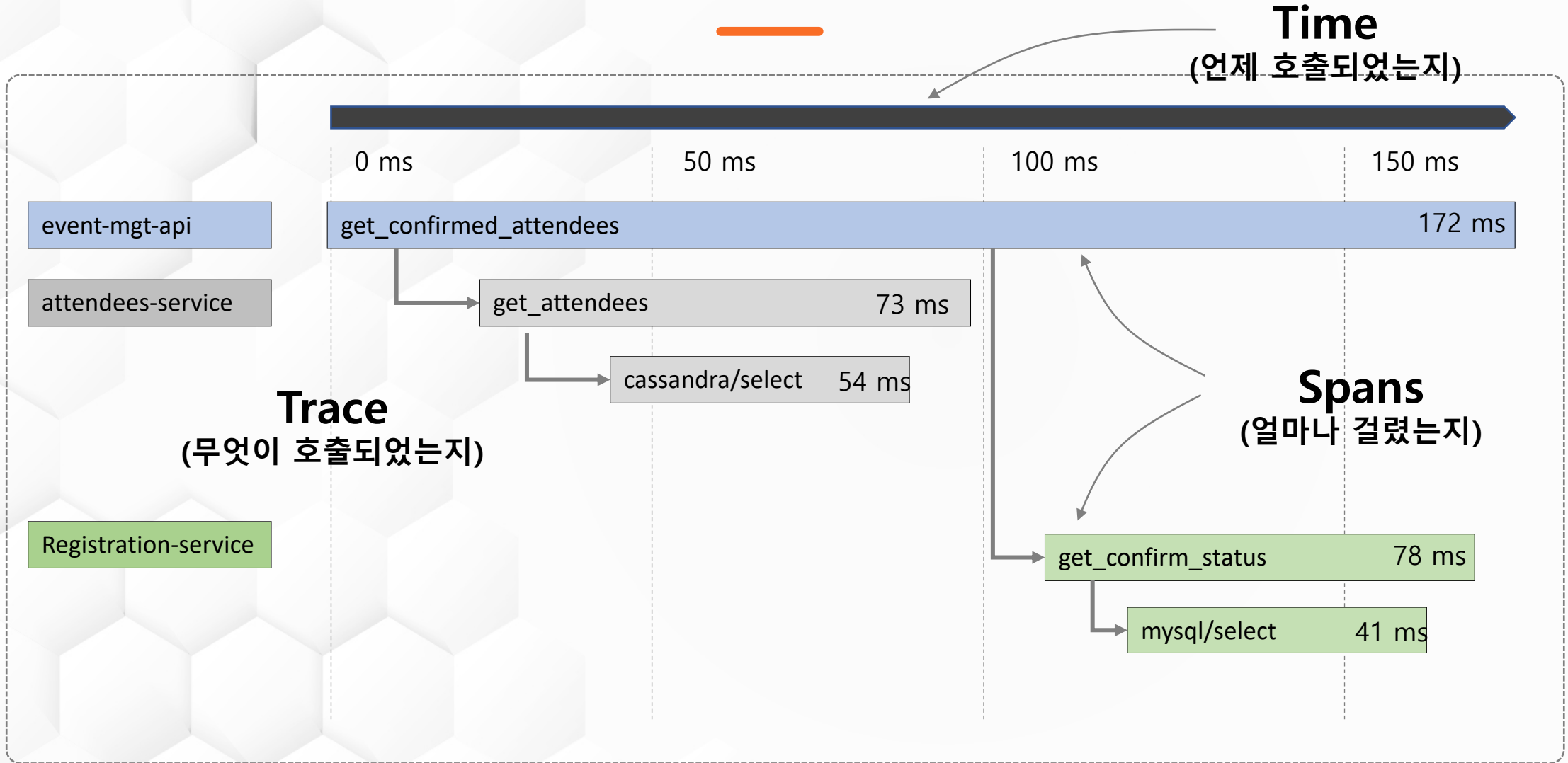
추적 가능한가?

requestID	ec667cb45
-----------	-----------

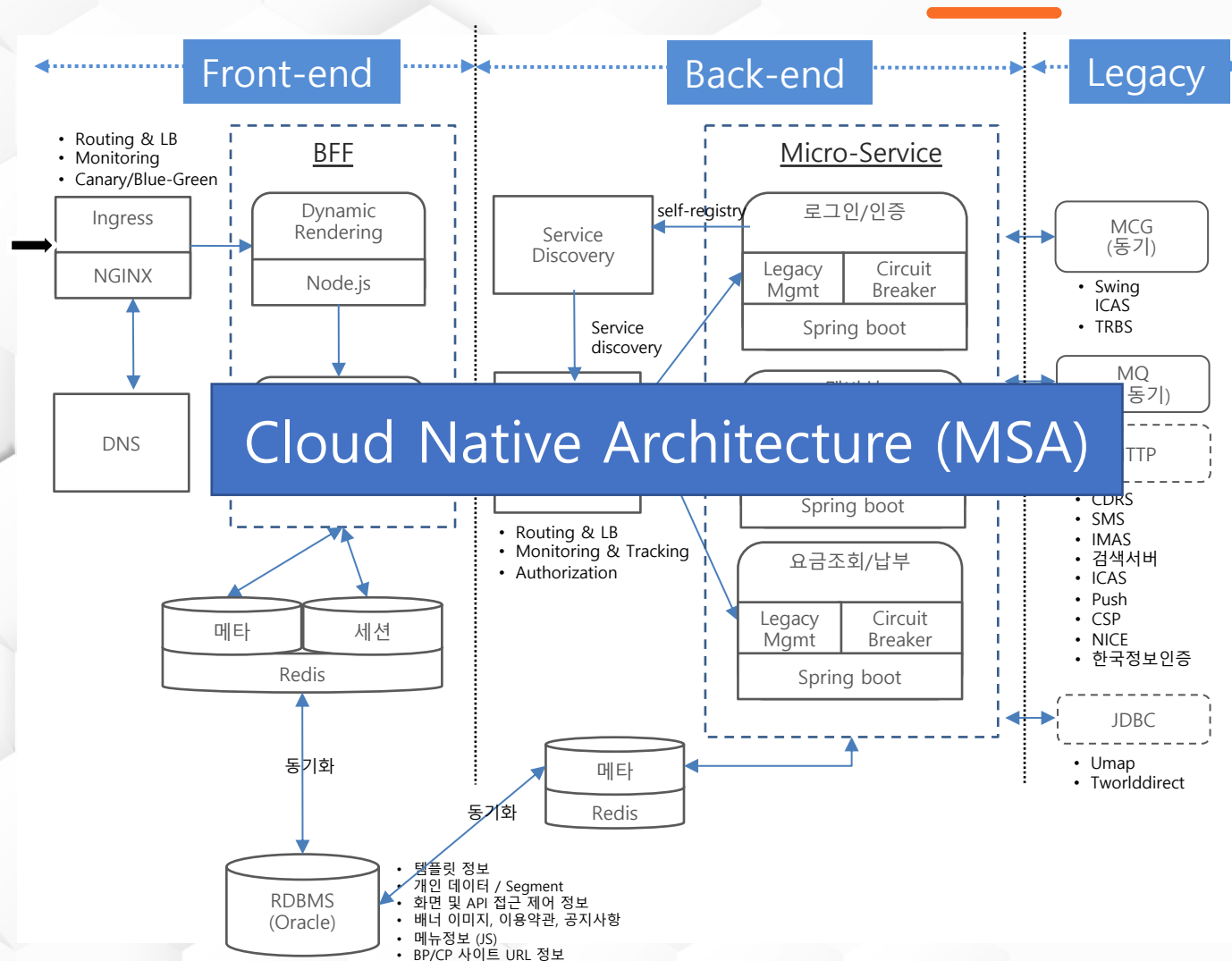
또다른 정보는?

....	...
------	-----

Tracing



[사례] MSA 기반 B2C 모바일 서비스 시스템 구축



3 Pillar Observability Approach



[사례] MSA 기반 B2C 모바일 서비스 시스템 구축

구분	세부 환경	고려할 점
Metric	Bare-metal	기존 물리 장비 모니터링 환경 유지
	Container 관리 Resource	K8S 관리영역내의 리소스 Metric 정의/대시보드 구성
	Container App Resource	필요 Metric 정의 및 대시보드 구성 (Pod 단위별 리소스, Redis 리소스 등)
Trace	Container 기반 APM	Multi Layer, Multi Transaction 흐름 Tracking 가능
Log	Application Log	DISK IO/Network IO 서비스 영향 없는 수집방식 대량의 애플리케이션 에러로그 검색속도 고려 애플리케이션 에러 코드별 발생빈도 Rule/Alerting

Cloud Service Observability

- 1 APM (Microservice)**

 - 다양한 APM 솔루션
 - 연계 모니터링 서비스 Integration
- 2 로그 대시보드**

 - 로그 표준 정의 (Appl, Sys, Container)
 - 로그정보 보안요건 검토
- 3 컨테이너 모니터링**

 - 다양한 CSP 서비스 수용성/확장성
 - 관리형 서비스 Metric 수집
- 4 Cloud Infra 모니터링**

 - Infra 표준 통합 Dashboard 제공
 - 대량의 Infra 설정 통합 관리

SKC&C Case

Cloud Observability Platform

Metric

통합된 모니터링



절차

수집

저장

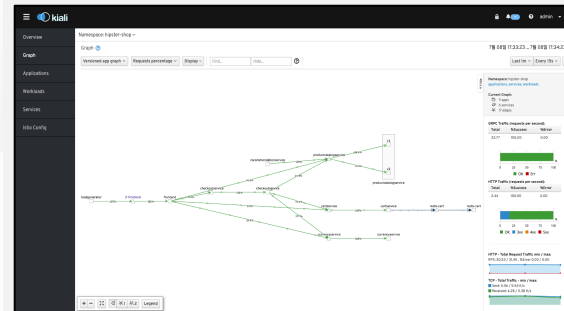
시각화

알람

- All-in-One으로 제공되는 통합모니터링 서비스를 통하여 Kubernetes, 클러스터에서 운영중인 Application 실행상태를 확인할 수 있습니다.
- Prometheus, Grafana기반의 모니터링 시스템을 클러스터의 모든 리소스에 대해 약 1,400여개의 매트릭을 수집하고 있으며, 클러스터 운영을 위한 기본알림과 기본 대시보드를 제공하여 별도의 설정없이 바로 모니터링 할 수 있습니다.

Tracing

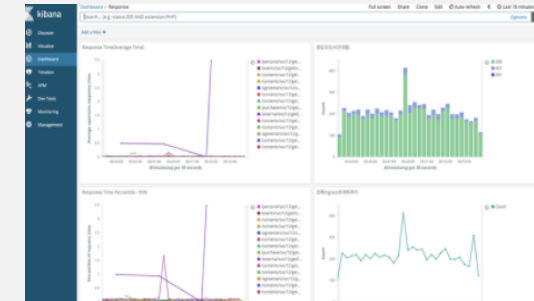
MSA Tracing 분석



- MSA Service간 처리.응답의 효율적 관리
 - Multi-Cluster 기반 분산된 서비스관리
 - MSA Application의 안정적인관리 서비스 (Response Time, Flow Percentage 등)
- Multi-Cluster 기반의 Traffic Routing
 - Traffic Routing 변화를 통한 서비스관리
 - MSA간 Flow Visualization 서비스관리

Logging

All-in-One 로깅



- All-in-One으로 제공되는 로깅 서비스를 통해 Kubernetes 클러스터에서 운영중인 Application 로그를 수집 및 조회할 수 있습니다.
- Elasticsearch, Kibana, FluentD 기반의 로깅시스템은 클러스터를 3개 데이터노드로 구성하여고가용성을 보장하며, 네임스페이스별로 인덱스를 구분하여 로그를 편리하게 조회/검색할 수 있는 기능 제공



감사합니다