Microservice & Immutable Infrastructure

엔터프라이즈 시스템/네트워크 운영자 대상

(for IT Pros and System Administrators)

2018. 09.

안종석
james@jslab.kr

- 사용 유효기간을 2019년 3월 까지로 권합니다.
- 실습 교재는 별도 입니다.

JS Lab

목차

Community for KOREN AI Network Lab

james@jslab.kr

JS Lab

목차

A. **마이크로서비스**

B. **컨테이너** (별도 교재)

C. **Immutable Infrastructure** (별도 교재)

D. **실습** (별도 교재)

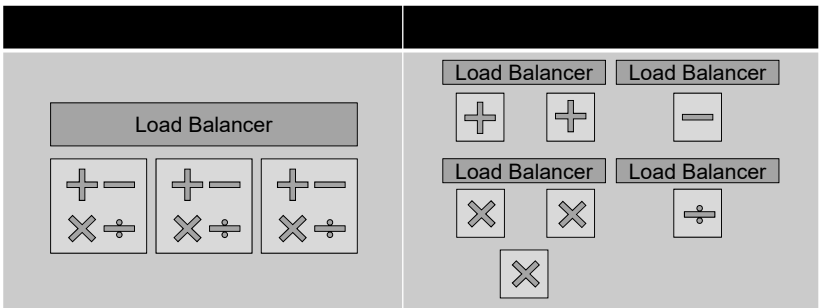
Community for KOREN AI Network Lab
james@jslab.kr

A. 마이크로서비스(Microservices)

❖ **What Are Microservices?**

- **마이크로서비스란 아키텍처이자 소프트웨어 작성을 위한 하나의 접근 방식으로, 애플리케이션을 상호 독립적인 최소 규모 구성 요소로 분할하여 마이크로서비스간 연결**(연결 예: http, RESTFUL web service, 메세지큐)
- **모든 요소를 하나의 애플리케이션에 구축하는 전통적인 모놀리스식(Monolithic) 접근 방식 대신 마이크로서비스에서는 모든 요소가 분리되고 연동되어 동일한 태스크를 완수 이러한 각각의 구성 요소 또는 프로세스가 마이크로서비스**

Kocher, Parminder Singh. Microservices and Containers. Pearson Education



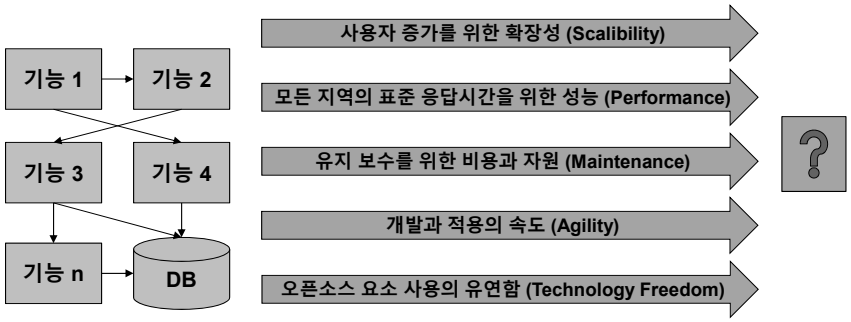
Community for KOREN AI Network Lab
james@jslab.kr

JS Lab

A. 마이크로서비스(Microservices)

❖ 현재의 애플리케이션 아키텍처 문제 (예)

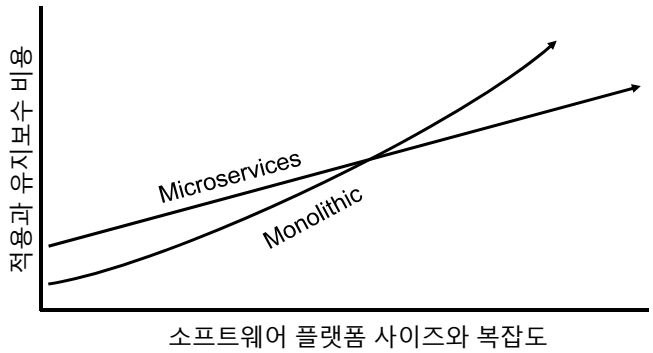
- 40명 이상의 개발자
- 글로벌 사용자
- 100,000 이상의 사용자
- 100.000 라인 이상의 코드



JS Lab

A. 마이크로서비스(Microservices)

❖ 비용



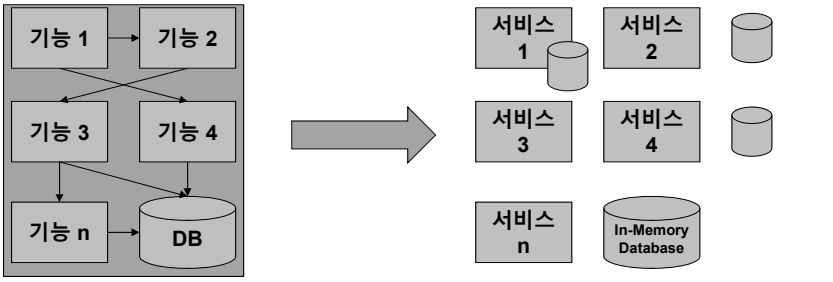
JS Lab

A. 마이크로서비스(Microservices)

❖ 마이크로서비스 전환 시 추가 기능 고려

- 비즈니스 기능을 위한 서비스 연결
- 스탠드얼론 and/or 서비스의 부분 적용
- 비동기 통신
- 성능 개선을 위한 서비스의 플랫폼 요소 교체 (프로그램 언어, 데이터베이스 등)
- 일정하지 않은 확장 요구 **CI/CD** (Continuous integration and Continuous delivery)
- 각 엔지니어링 팀은 비즈니스 영역의 이해하고 책임을 소유

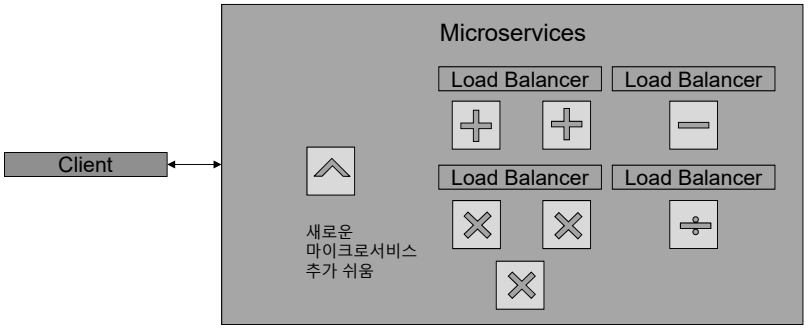
❖ 마이크로서비스 전환 시 조직 문화의 변화와 운영 프로세스 변화



A. 마이크로서비스(Microservices)

❖ 마이크로서비스의 장점

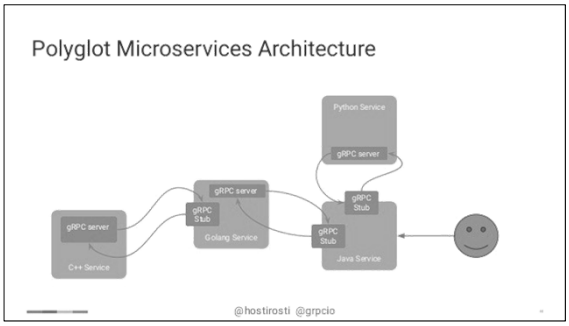
- 단순성(Simplicity)
- 확장성(Scalability)
- 적용(Continuous delivery)
- 낮은 의존성과 더 많은 자유
- 장애 확인
- 데이터 분리와 분산화
- 다양한 선택



A. 마이크로서비스(Microservices)

❖ gRPC 사용 마이크로서비스

- **gRPC is faster than REST. gRPC uses HTTP2 by default**
- **gRPC defines relationship between client and server and enforces strict rules of communication between them.** (In REST calls, the request and response are totally de-coupled)
- **gRPC supports useful additions like standard error responses and meta data.**

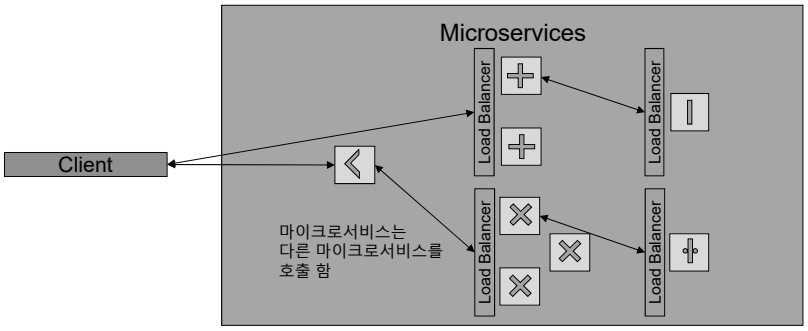


https://medium.com/@akshitjain_74512/inter-service-communication-with-grpc-d815a561e3a1

A. 마이크로서비스(Microservices)

❖ 마이크로서비스의 단점

- **트러블슈팅의 복잡성**
- **Latency 증가**
- **운영의 복잡성**
- **버전 제어**



A. 마이크로서비스(Microservices)

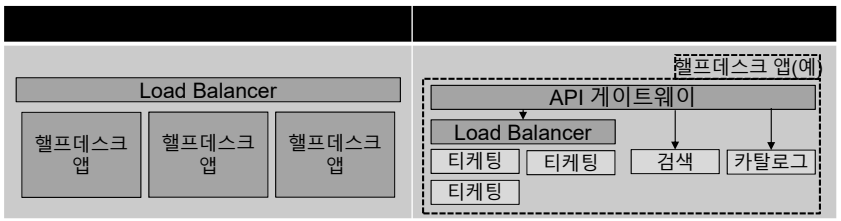
❖ 조직의 Learning Curve

- 단일 마이크로서비스 (Standalone Microservice) 많이 필요시
- 과도한 의존성으로 시간이 많이 필요하고 코드의 품질이 낮아질 때
- 한가지 요소로 애플리케이션 장애 시

❖ 마이크로서비스 전환 시 추가 기능 고려

- 비즈니스 기능을 위한 서비스 연결
- 스탠드얼론 and/or 서비스의 부분 적용
- 각 엔지니어링 팀은 비즈니스 영역의 이해하고 책임을 소유

❖ 마이크로서비스 전환 시 조직 문화의 변화와 운영 프로세스 변화



A. 마이크로서비스(Microservices)

❖ 유지보수

- 기존 클라이언트 적용 지원
- 장애 고려 (적절한 error code 생성 등)
- 모니터링 (가용성이나 오케스트레이션 등을 지원하는 도구사용)
- 큐

Community for KOREN AI Network Lab
james@jslab.kr

JS Lab

A. 마이크로서비스(Microservices)

❖ **Discovery Service**

- **Client는 기존에는 1개의 monolithic app을 하였으나 MSA에서는 동시에 여러 개의 서비스를 호출해야 함**
- **Client는 서비스들의 위치를 알아야 함 (Host/IP/Port 등)**
- **Client가 마이크로서비스 호출시 제공하는 entry point 실시간 위치를 하드코드(Hard code)보다는 유연한 방법으로 현재의 위치를 제공해야 함**

❖ **API Gateway**

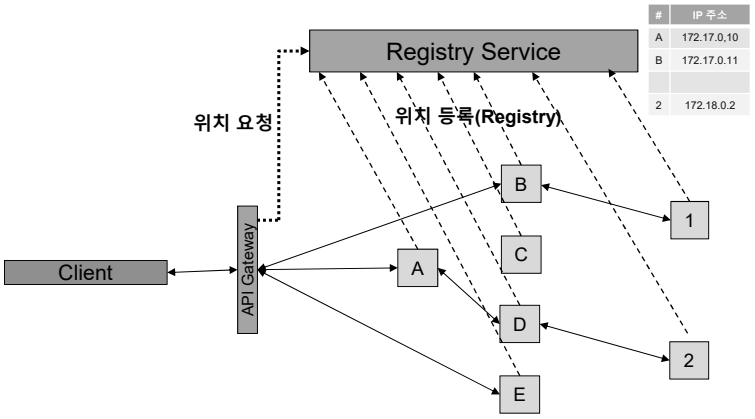
- **모든 호출에 대한 1개의 entry point가 필요**
- **내부의 복잡성을 숨김**
- **마이크로 서비스 변화/결합/구분/추가/제거 유연함**
- **Client와 Application사이의 왕복 단순화로 효율성 증대**

Community for KOREN AI Network Lab
james@jslab.kr

A. 마이크로서비스(Microservices)

❖ **Service registry**

- **API Gateway는 수천개의 모든 서비스에 대한 IP 주소를 알아야 하며 이의 DB 필요**
- **Registry 데이터의 안정성을 위해 오픈소스 사용 가능**(Consul이나 SkyDNS)

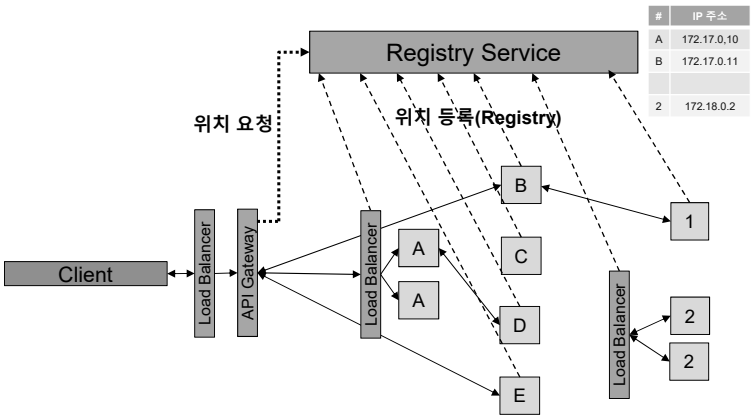


Community for KOREN AI Network Lab
james@jslab.kr

A. 마이크로서비스(Microservices)

❖ 로드밸런서 추가시 Service registry

- API Gateway는 수천개의 모든 서비스에 대한 IP 주소를 알아야 하며 이의 DB 필요
- Registry 데이터의 안정성을 위해 오픈소스 사용 가능(Consul이나 SkyDNS)



A. 마이크로서비스(Microservices)

❖ 적용 기술 선택

- Independency
- Source Control
- Environment
- Failsafe
- Reuse
- Tagging (로그 관리 예: Splunk나 ELK or 'Elasticsearch, Logstash, Kibana')

❖ 운영 기술 선택

- 모니터링: 인프라의 모든 요소를 모니터하며 라이브 대쉬보드와 이슈 팝업 사용
- 온디맨드 확장성: 수동 또는 자동화
- API 노출: Netflix의 경우 Zuul 오픈소스로 공개
- 서킷 브레이커 Circuit Breaker): 모든 시비스에 대해 요청모드를 요구하여 장애를 확인하며, 복수의 장애시 특정 서비스를 차단(Break the circuit)함

A. 마이크로서비스(Microservices)

❖ 성공 사례

- 아마존
- Netflix
- 우아한 형제들 (배달의 민족) Spring Cloud zuul 적용
- 11번가 Spring Cloud 적용
- SK C&C C-NAPS 방법론 개발한가지 요소로 애플리케이션 장애 시

❖ 마이크로서비스 사례

- http://channy.creation.net/articles/microservices-by-james_lewes-martin_fowler (마틴 파울러의 마이크로 서비스 정의)
- https://www.youtube.com/watch?v=OczG5FQIcXw (넷플릭스 마이크로 서비스 가이드 – 혼돈의 제왕)
- http://woowabros.github.io/r&d/2017/06/13/apigateway.html (우아한 형제들 spring cloud zuul 적용기)
- https://www.popit.kr/why-microservice/ (마이크로 서비스의 장단점)
- SK C&C C-NAPS Overview

Community for KOREN AI Network Lab
james@jslab.kr

5. Q&A

Community for KOREN AI Network Lab
james@jslab.kr