교육 결과 보고서

[Online] Spring Cloud를 활용한 MSA 기초

1. Monolithic Architecture

마이크로 서비스 아키텍쳐의 반대되는 개념으로 전통의 아키텍쳐를 지칭함.

하나의 서비스 또는 애플리케이션이 하나의 거대한 아키텍쳐를 가질 때, 모놀리틱 아키텍쳐라 함.

* 1. 장점

개발 초기에 단순한 아키텍쳐 구조로 개발에 용이함.

배포가 단순함.

Scale-out 이 단순함.

* 1. 단점

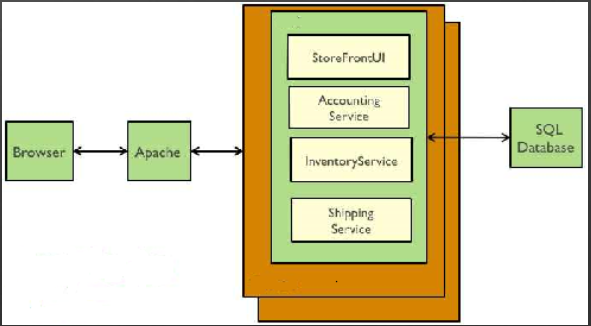
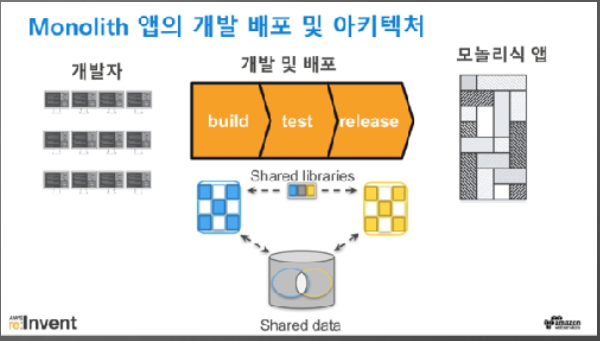
모든 서비스가 한 개의 코드 베이스를 공유하므로 IDE 입장에서 무겁다.

애플리케이션 시작이 오래 걸린다.

기술 스택을 바꾸기 어렵다.

서비스 요소 간 높은 결합도를 갖는다.

코드베이스의 책임 한계와 소유권이 불투명 하다.



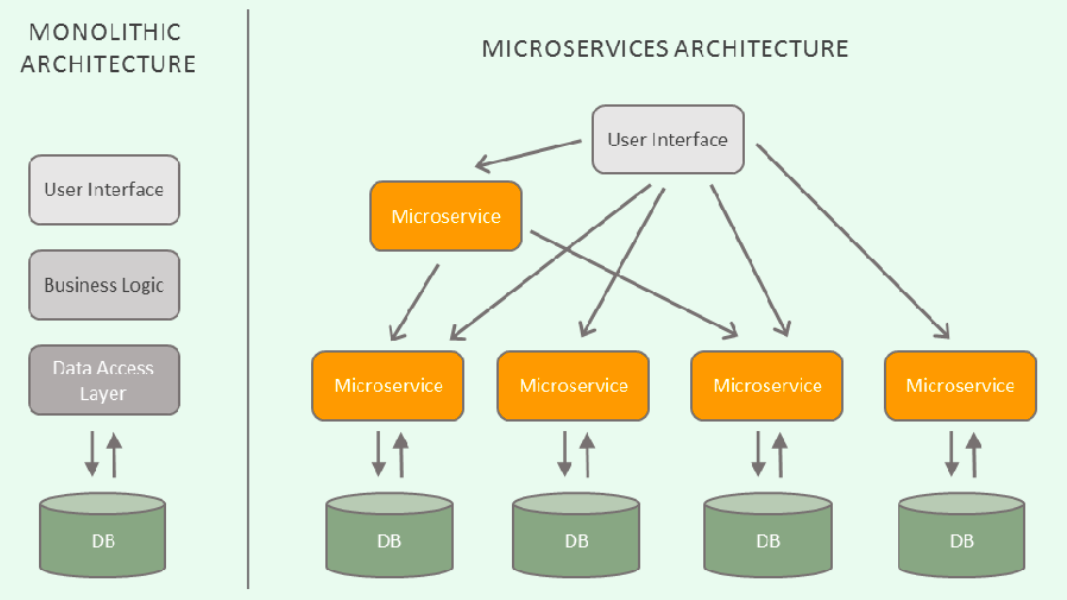
2. MSA (Micro Service Architecture)

시스템을 여러 개의 독립된 서비스로 나누고, 이 서비스를 조합함으로서 기능을 제공하는

아키텍쳐 디자인 패턴.

공식적인 정의는 없으며, 아래 내용의 공감대가 있음.

* Network를 통해 각 서비스 간 조합(보통 HTTP 이용)
* 독립된 배포 단위
* 개별 서비스는 쉽게 교체 가능
* 개별 서비스는 기능 중심으로 구성
* 개별 서비스에 적합한 프로그래밍 언어, 데이터베이스, 환경 이용
* 서비스는 크기가 작고, 상황에 따라 경계를 정하고, 자율적으로 개발되고, 독립적으로 배포되고, 분산되고, 자동화 된 프로세스로 구축 되고 배포된다.



3. Cloud Native

핵심은 애플리케이션을 어떻게 만들고 배포하는지에 있으며 위치는 중요하지 않다는 것.

클라우드의 이점을 최대로 활용할 수 있도록 애플리케이션을 구축하고 실행하는 방식으로

생산성, 비즈니스 민첩성, 확장성, 가용성을 높이고 비용을 절감 할 수 있음.

3.1 주요 요소

\* DevOps

애플리케이션 개발-운영 간 협업 프로세스를 자동화 하는 것으로

애플리케이션의 개발과 개선 속도를 빠르게 함.

\* CI/CD

CI (Continuous Integration) : 지속적인 통합, 변경 사항을 자동으로 테스트하고 통과된 코드

를 통합하여 저장.

CD(Continuous Deployment) : 지속적인 배포, 변경 사항은 테스트를 거쳐 리포지토리에

업로드 되고 서비스 배포로 릴리즈 까지 자동화 하는 것

\* 컨테이너 기반 인프라

애플리케이션을 구동할 수 있는 컴퓨팅 작업을 패키징하여 가상화 한 것으로

다양한 환경으로 이동 가능하도록 하는 기술.

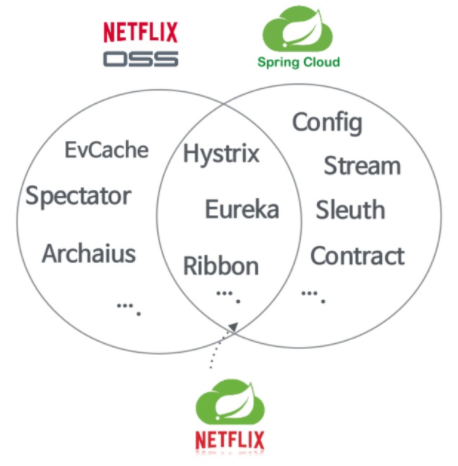
\* Micro Service

애플리케이션을 구성하는 서비스들을 독립적인 작은 단위로 분해하여 구축하고 네트워크를

이용하여 개별 서비스 간 통신하는 아키텍쳐

4. Spring Cloud

Spring Boot 기반의 MSA 구축에 특화된 라이브러리의 집합.



4.1 Hystrix

서비스 장애 상황 (네트워크 연결 불가, 서버 다운 등) 에 대응하기 위한 라이브러리.

\* 다른 서비스의 실패에 따른 영향 방지

\* 분산 시스템의 복잡한 연쇄 실패 방지

\* 빠르게 실패 하고, 빠르게 복구

\* Fallback 과 Gracefully degrade

\* real-time에 유사한 모니터링과 알람

4.2 Ribbon

클라이언트 사이드 로드 밸런싱 라이브러리

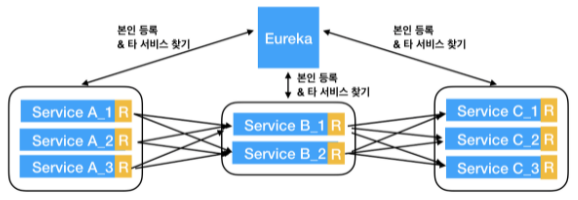
주어진 서버 목록에 대해 Load Balancing 및 Retry 기능 수행.

4.3 Eureka

Cloud 시스템에서 서비스의 로드밸런싱과 실패처리등을 유연하게 가져가기 위해

각 서비스들의 IP/Port/InstanceId 를 가지고 있는 REST 기반의 미들웨어 서버.

마이크로 서비스기반 아키텍쳐에서 Service Discovery 역할을 수행함.



4.4 Spring Cloud Feign

Interface 선언을 통해 자동으로 HTTP Client 를 만들어 주는 Declarative HTTP Client.

Ribbon, Eureka, Hystrix 와 통합 되어 있음.

* Http Client
* Eureka 타겟 서버 주소 획득
* Ribbon 을 통한 Client-Side Load Balancing
* Hystrix 를 통한 메소드 별 Circuit Breaker

4.5 Spring Cloud Zuul

API Gateway,

클라이언트와 백엔스 서버 사이의 출입문

라우팅, 필터링, API 변환, 클라이언트 어댑터 API, 서비스 프록시

횡단 관심사 cross-service concerns

* 보안, 인증, 인가
* 일정량 이상의 요청 제한
* 계측

