

아키텍처링 특강

이스트소프트 백엔드 개발자 양성 과정 오르미 1기

강사소개



계성혁

- ▶ 현) LG전자 데이터 엔지니어
- ▶ 전) (주) 더블에이치 서버 엔지니어 인턴
- ▶ 아주대학교 소프트웨어학과 졸업
- ▶ KAIST 소프트웨어대학원 졸업 (지도교수: 강지훈)

강의 시간표

시간	학습 내용
09:00 ~ 09:50	강의 소개 및 아키텍처링의 필요성
10:00 ~ 10:50	모니터링 및 아키텍처링 기본 지표 소개
11:00 ~ 11:50	컴퓨팅 계층의 아키텍처링
13:00 ~ 13:50	저장소 계층의 아키텍처링
14:00 ~ 14:50	네트워크 계층의 아키텍처링
15:00 ~ 15:50	아키텍처 확장을 위한 데이터 모델링
16:00 ~ 16:50	실전 프로젝트 아키텍처링 (SNS 서비스)
17:00 ~ 17:50	프로젝트 아키텍처링 해설

왜 (인프라) 아키텍처링이 필요한가?

High availability (HA) is a characteristic of a system which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period.

Modernization has resulted in an increased reliance on these systems. For example, hospitals and data centers require high availability of their systems to perform routine daily activities. Availability refers to the ability of the user community to obtain a service or good, access the system, whether to submit new work, update or alter existing work, or collect the results of previous work. If a user cannot access the system, it is – from the user's point of view – unavailable.[1] Generally, the term downtime is used to refer to periods when a system is unavailable.

Source: [High Availability - Wikipedia](#)

왜 (인프라) 아키텍처링이 필요한가?

실제 서비스는 애플리케이션 코드 그 자체만으로 동작하는 것이 아닙니다

- ▶ 애플리케이션 코드가 실행될 서버 환경
- ▶ 애플리케이션 코드가 사용할 데이터 저장소
- ▶ 서비스에 접근할 수 있도록 하는 네트워크 환경 등

애플리케이션 자체의 성능 최적화도 필요하지만, 서비스의 규모에 맞게 확장하는 것이 필요합니다.

- ▶ 이 때 잘못된 인프라 설계는 서비스의 확장을 저해합니다.

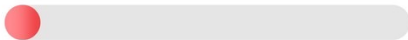
예를 들어, 티켓팅

예매하러 왔는데
조금만 기다려주세요요~



나의 대기순서

172573



현재 접속 인원이 많아 대기중입니다.
잠시만 기다려주시면 예매하기 페이지로 연결됩니다.

❗ 새로고침 하거나 재접속 하시면
대기순서가 초기화되어 대기시간이 더 길어집니다.

만약 티켓팅 서버에 장애가 난다면?

- ▶ 일단 장애 복구를 위한 개발자들의
피땀눈물이 필요합니다(...)
- ▶ 그러나 고객들은 그 복구 시간을 끝까지
기다려주지 않습니다.

장애는 곧 서비스의 실적 하락으로
이어집니다! (예. 매출)

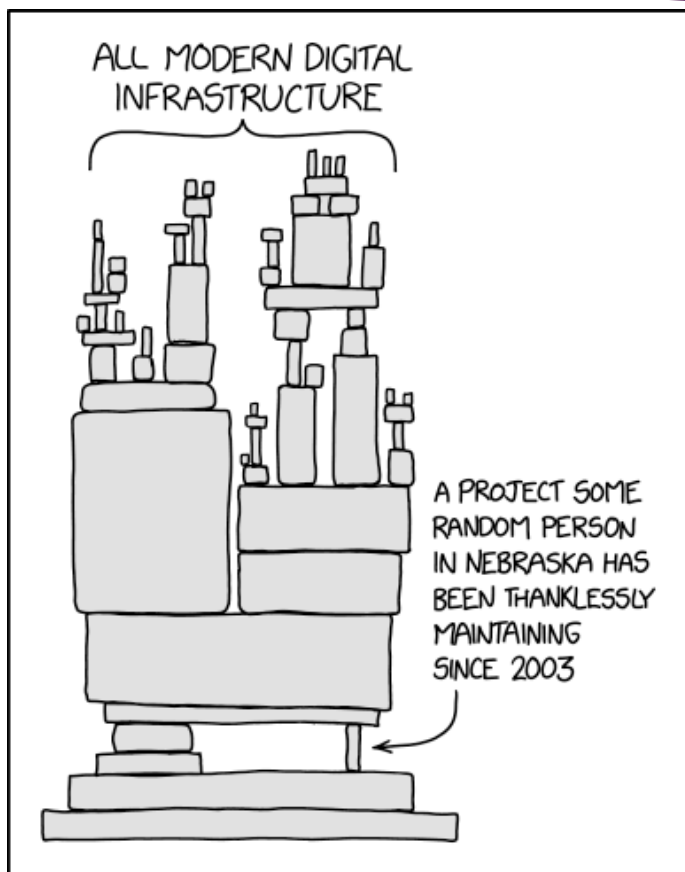
아키텍처링의 예시

한 회사에서 Amazon API Gateway, AWS Lambda 함수, Amazon Cognito, Amazon DynamoDB를 사용하는 서버리스 모바일 앱을 실행합니다. 트래픽이 급증하는 기간에는 사용자가 간헐적인 시스템 장애를 보고합니다. API Gateway API 엔드포인트가 HTTP 상태 코드 502(잘못된 게이트웨이) 오류를 유효한 요청에 반환하고 있습니다.

어떤 솔루션으로 이 문제를 해결할 수 있습니까?

- A. Lambda 함수에 대한 동시성 할당량을 늘립니다. ConcurrentExecutions 지표가 할당량에 다다를 때 경고 알림을 보내도록 Amazon CloudWatch 를 구성합니다.
- B. API Gateway API 엔드포인트에서 초당 트랜잭션 할당량에 대한 경고 알림을 구성합니다. 할당량에 도달하면 할당량을 늘리는 Lambda 함수를 생성합니다.
- C. 사용자를 여러 AWS 리전의 Amazon Cognito 사용자 풀로 분할하여 사용자 인증 대기시간을 줄입니다.
- D. DynamoDB 의 강력한 읽기 일관성을 바탕으로 클라이언트 애플리케이션이 항상 최신 데이터를 수신하도록 합니다

아키텍처링은 답이 없습니다.



아무리 코드를 지금 당장 잘 짜도, 완성된 순간부터 레거시 시스템이 됩니다.

- ▶ 레거시 시스템을 뒤엎고 새로 개발하는 것은 큰 용기가 필요합니다.
- ▶ 그렇기 때문에 아키텍처링의 핵심은 문제를 확인하고, 원인을 파악하여, 현 상황에 맞는 해결 전략을 수립하는 것입니다.

오늘 특강 부담 갖지 마세요!

사실 아키텍처링을 하루에 다 다루기는 어렵습니다.

- ▶ 대부분은 직접 서비스 올려서 운영하고, 개발자로 일하면서 습득하게 되는 부분들이 많기 때문입니다.
- ▶ 오늘 배운 내용들은 키워드만 머리에 담아두고, 미래에 필요할 때 꺼내서 사용해 보세요.



강의 진행 방식 소개

1, 2교시는 제가 먼저 설명을 진행합니다.

3~6교시는 제시한 문제 상황에 대해 팀별로 해결 방안을 토론해봅니다.

- ▶ 토론한 내용을 발표하고

- ▶ 모범 답안과 핵심 지식들을 설명하는 방식으로 진행합니다.

7교시는 프로젝트 아키텍처링 실습을 팀별로 진행하고

이 결과를 8교시에 서로 공유하고 모범 답안을 제시하면서 마무리합니다.

강의 시간표

시간	학습 내용
09:00 ~ 09:50	강의 소개 및 아키텍처링의 필요성
10:00 ~ 10:50	모니터링 및 아키텍처링 기본 지표 소개
11:00 ~ 11:50	컴퓨팅 계층의 아키텍처링
13:00 ~ 13:50	저장소 계층의 아키텍처링
14:00 ~ 14:50	네트워크 계층의 아키텍처링
15:00 ~ 15:50	아키텍처 확장을 위한 데이터 모델링
16:00 ~ 16:50	실전 프로젝트 아키텍처링 (SNS 서비스)
17:00 ~ 17:50	프로젝트 아키텍처링 해설



쉬는 시간!

10시에 만나요

모니터링과 기본 지표 소개

들어가기에 앞서!

설계한 아키텍처의 내용을 그려내기 위해 도구가 필요합니다.

- ▶ 다양한 도구들이 있지만 우리는 draw.io를 사용할 것입니다.
- ▶ <https://app.diagrams.net>

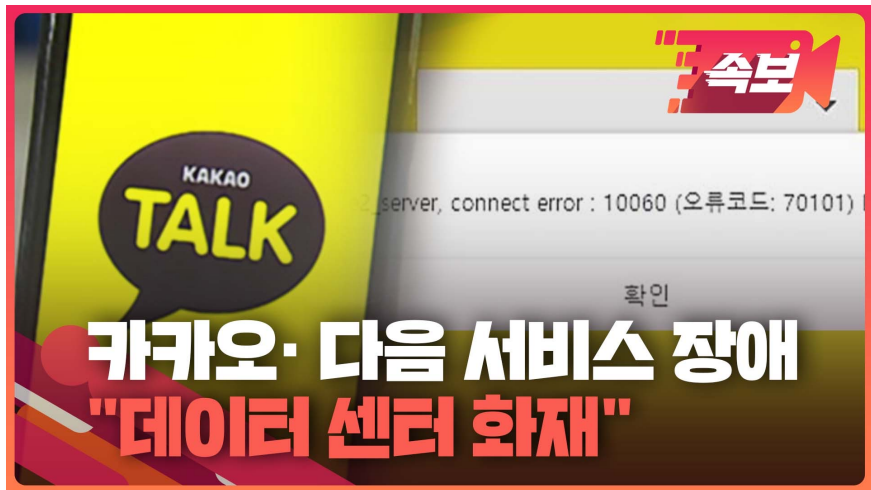
모니터링

서비스가 운영되고 있는 각종 상황들을 확인하는 행위 전반

모니터링의 필요성?

- ▶ 서비스의 장애 상황을 (사용자의 리포트를 통해서 아는 것이 아니라) 선제적으로 발견하여 대응할 수 있도록 합니다.

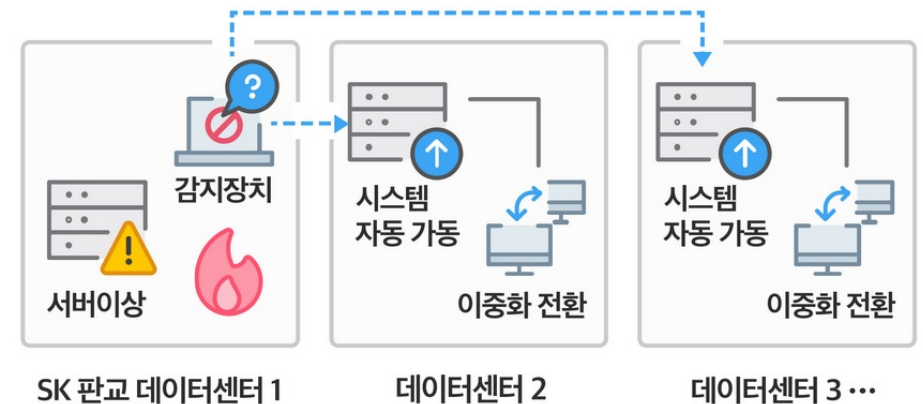
작년 10월...



2 이중화된 시스템의 자동화를 위한 모니터링 시스템 구축 및 자원확보의 부족

이중화 전환과 트래픽 제어가 자동으로 매끄럽게 이뤄지지 않고, 수동으로 대응할 수밖에 없는 상황들이 존재했어요. 데이터센터 간 이중화가 미흡한 상태에서, 모니터링 시스템이 충분히 갖춰져 있지 않아 수동으로 이중화 전환을 해야 했던 케이스들이 발견됐습니다. 여러 시스템에 동시다발적으로 문제가 생기는데, 의존하는 다른 시스템의 문제를 인지하지 못해 자동 전환에 실패하는 경우죠.

또한 전체 시스템의 이중화 수준은 유발할 수 있습니다. 개별 부서나 사 체계적인 이중화를 준비했어야 했



Source: [속보영상] 카카오·다음 서비스 장애... "데이터 센터 화재" | KBS 뉴스
우리가 부족했던 이유 | 카카오

지표 (Metric)

*In software engineering and development, a **software metric** is a standard of measure of a degree to which a software system or process possesses some property.*

지표들은 서비스와 관련된 다양한 양상들을 보여줍니다.

- ▶ 호스트 지표: CPU, 메모리, 디스크, 네트워크 I/O 등 시스템 관련 수치
- ▶ 종합(Aggregated) 지표: 캐시 적중율(Hit Ratio), DB 커넥션 갯수
- ▶ 비즈니스 지표: DAU/MAU, (구매) 전환율

지표들은 어떻게 확인할까?

리눅스에서 자원 관련 지표들은 다양한 명령어로 확인할 수 있습니다.

- ▶ 예) ps, top(htop)

클라우드 서비스에서는 자체 모니터링 체계로 확인할 수 있습니다.

- ▶ 예) AWS CloudWatch

비즈니스 관점의 지표들은 별도의 도구들을 활용해 확인할 수 있습니다.

- ▶ 예) Google Analytics



쉬는 시간!

11시에 만나요

컴퓨팅 계층의 아키텍처링

문제 상황

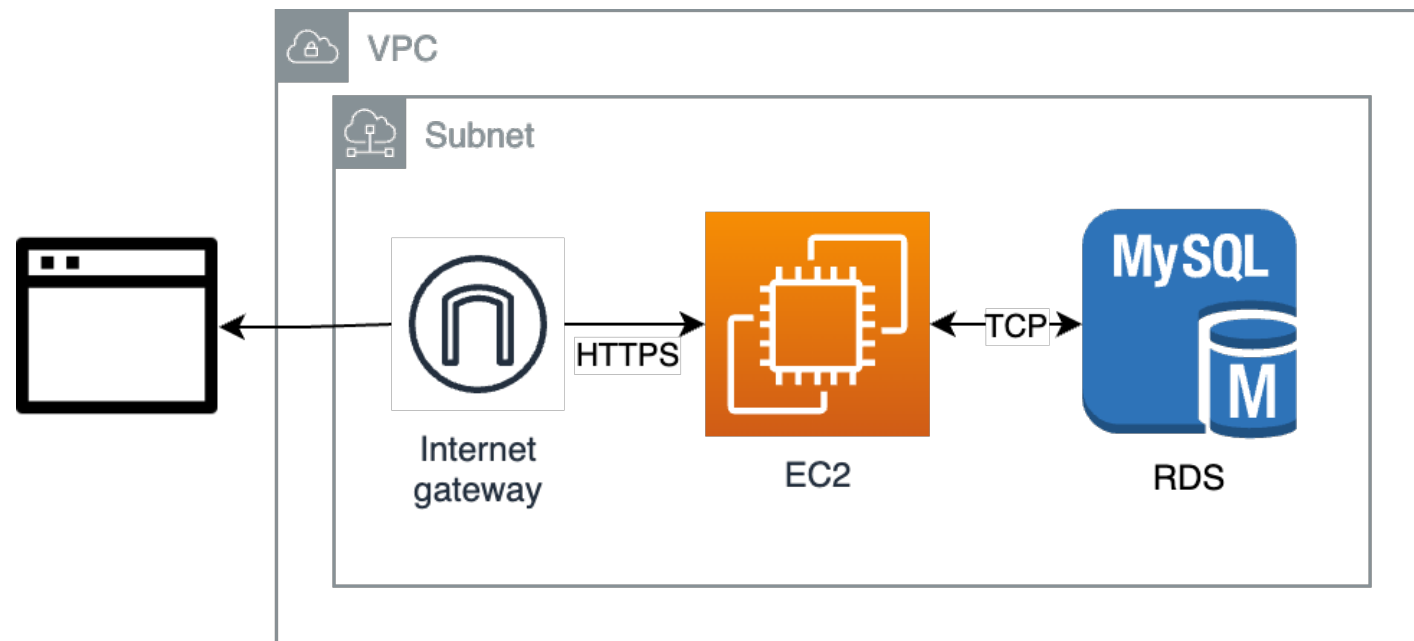
우리가 쇼핑몰 서비스를 운영하고 있다고 가정해봅시다.

최근 들어 이 쇼핑몰 서비스에서 평일 오후 2시에 특가 상품 판매를 시작했는데, 이용자들이 접속을 못해 상품 구매를 못하고 있다는 불만이 접수되고 있습니다.

- ▶ 사용자들이 Time-out 페이지를 마주하고 있습니다.
- ▶ 관리자가 상황 수습을 위해 서버에 원격 접속(SSH)를 시도해봤지만, 접속이 불가능하여 대응이 늦어지고 있습니다.
- ▶ 모니터링 지표로 CPU 점유율, 서버 애플리케이션 상 응답 지연 시간, 그리고 Time-out 횟수가 증가한 것을 확인했습니다.

5분 동안 자유롭게 방법을 생각해 보세요!

10시 10분까지 진행합니다.





쉬는 시간!

1시에 만나요. 식사 맛있게 하세요~

저장소 계층의 아키텍처링

문제 상황

쇼핑몰 서비스의 사용자들이 서버 접속을 할 수 있게 됐지만, 간헐적인 오류와 지연을 겪는다는 불만이 추가로 접수되고 있습니다.

- ▶ 메인 페이지의 실시간 검색 순위에 표시되는 상품 페이지는 금방 나타나지만, 그 이외의 상품들은 늦게 표시된다고 합니다.
- ▶ 오후 2시에 열리는 "특가 상품" 페이지에 접속할 때 "DB Connection Limit Exceeded"라는 메시지가 발생한다고 합니다.
- ▶ 모니터링 지표에서 DB Free Memory 부족, DB Select Statement Latency 증가하고 있다는 사실을 확인했습니다.



5분 동안 자유롭게 방법을 생각해보세요!

2시 15분까지 진행합니다.



쉬는 시간!

2시에 만나요

네트워크 계층의 아키텍처링

문제 상황

쇼핑몰 서비스가 한류의 인기에 힘입어 일본어로도 운영되어 일본 사용자들이 들어오기 시작했습니다.

하지만 어느 날 AWS 서울 리전 전체에 장애가 발생하여 국내 사용자 뿐만 아니라, 일본 사용자도 84분 동안 접속을 못하는 상황이 발생했습니다(!)

- ▶ 다행히 장애 시간동안의 인프라 비용은 AWS에서 보상해줬지만, 84분 동안 상품 판매가 없어서 손실을 입었습니다.
- ▶ 이러한 장애 상황에도 사용자들이 정상적으로 접속하여 상품 판매가 이뤄질 수 있도록 아키텍처를 개선하고자 합니다.

사전 지식 (리전, Region)

클라우드 서비스 제공자(CSP)들은 각자의 내결함성 확보와 네트워크 지연 시간을 감소 시키기 위해 자신들의 데이터센터를 다양한 지역(Region)에 분산하여 운영합니다.

- ▶ 내결함성? 하나의 리전이 장애가 나더라도 서비스에 영향이 없도록 하는 것
- ▶ 네트워크 지연시간 감소? 사용자가 서비스를 사용하는데 가장 가까운 인프라에 접근하도록 하는 것!

리전은 다양한 가용 영역(Availability Zone, AZ)으로 구성되어 있으며, 이 또한 서로 격리되어 있어 영역별 장애에 대한 내결함성을 보장할 수 있습니다.



예) AWS 리전 현황

Source: [글로벌 인프라 리전 및 가용 영역](#)

사전 지식 (DNS)

DNS는 인터넷 상에 공개되어 있는 각종 자원들을 인간이 읽을 수 있는 형태로 접근할 수 있도록 해주는 체계입니다. (참고. [RFC 1035](#))

- ▶ 인터넷 상에 공개되어 있는 각종 자원들 IP 주소로 접근 가능 (예. [3.37.232.86](#))
- ▶ 인간이 읽을 수 있는 형태 인간이 사용하는 문자로 표현된 형태 (예. [www.estsoft.ai](#))

일부 DNS 서비스는 RFC의 표준을 확장해 다양한 요구사항을 수용할 수 있는 추가 기능을 제공합니다.

- ▶ 예) [라우팅 정책 선택 - Amazon Route 53](#)



5분 동안 자유롭게 방법을 생각해 보세요!

2시 15분까지 진행합니다.



쉬는 시간!

3시에 만나요.

아키텍처 확장을 위한 데이터 모델링

Source: 알렉스 쉬, 『가상 면접 사례로 배우는 대규모 시스템 설계 기초』 (인사이트, 2021), p127-p125.
마틴 클레프만, 『데이터 중심 애플리케이션 설계』 (위키북스, 2018), p199-281.

문제 상황

일본 서비스 오픈 이후 사용자 수가 폭발적으로 증가하면서 쇼핑몰 서비스가 호황을 이루고 있습니다. 하지만 일본 서비스 오픈 1달이 지난 지금, 서비스의 성능 저하에 대한 불만이 올라오기 시작했습니다. 일부는 서비스 오류에 대한 불만도 제기되고 있습니다.

- ▶ 서비스 오류 - A 사용자는 회원가입을 하고 로그인 했는데, B 사용자의 정보로 로그인이 되는 것을 확인했으며, 반대로 B 사용자는 회원가입 직후 로그인이 계속 실패된다고 합니다.
- ▶ 성능 저하 - 상품 검색을 할 때 결과를 불러오는 시간이 오래 걸립니다. 특히 검색 조건을 추가할수록 시간이 더 오래 걸린다고 합니다. (시간이 지날수록 더 많은 사용자로부터 문의가 들어왔습니다.)

(검색 엔진을 도입하는 것이 필요하지만, 이는 시간이 오래 걸려 현 단계에서는 제외합니다.)

문제 상황

애플리케이션 로그와 DBMS의 쿼리 로그를 확인해보니 A 사용자와 B 사용자는 완벽히 동시에 (1ms의 오차 없이!) 회원가입을 요청했던 것으로 확인되었으며, A 사용자의 가입 정보는 DB에 안 남아있는 것으로 확인했습니다.

- ▶ 이 오류는 한 사례이며, 유사한 오류들이 반복적으로 리포팅되고 있습니다.
- ▶ 의심해볼 것은 회원 정보 테이블의 Primary Key를 Auto Increment를 사용하지 않고 있던 것입니다.

"성능 저하" 건은 애플리케이션 디버깅 결과 병목 걸릴만한 지점이 발견되지 않았습니다. 단, 검색 결과 로딩에 사용되는 쿼리를 분석해보니 Table Full Scan이 발생하고 있었습니다.



5분 동안 자유롭게 방법을 생각해 보세요!

3시 15분까지 진행합니다.



쉬는 시간!

4시에 만나요.

프로젝트 아키텍처링

Source: 알렉스 쉬, 『가상 면접 사례로 배우는 대규모 시스템 설계 기초』 (인사이트, 2021), p183-p196.

실전 설계!

앞서 배웠던 개념들을 적용해서 실제 프로젝트 일부의 아키텍처를 세워봅시다.

- ▶ SNS 서비스의 “뉴스 피드 시스템”을 설계해봅시다.
- ▶ 뉴스 피드? 친구/팔로잉의 소식과, 그와 관련된 다른 소식들이 지속적으로 업데이트되어 나타나는 공간 (글, 사진, 영상, 링크 등)

요구사항

모바일 앱을 위한 시스템인가요? 아님 웹? 둘 다 지원해야 하나요?

- ▶ 둘 다 지원해야 합니다.

중요한 기능으로는 어떤 것이 있을까요?

- ▶ 사용자는 뉴스 피드 페이지에 새로운 스토리를 올릴 수 있어야 하고, 친구들이 올리는 스토리를 볼 수도 있어야 합니다.

뉴스 피드는 어떤 순서로 소식들이 표시가 되어야 하나요? 별도의 가중치가 있나요?

- ▶ 단순히 시간 흐름의 역순, 즉 최신 소식이 먼저 보이도록 설계하시죠

요구사항

한 명의 사용자는 최대 몇 명의 친구를 가질 수 있습니까?

▶ 5,000명입니다.

트래픽 규모는 어느 정도입니까?

▶ 매일 천만 명이 방문한다고 가정합니다.

Hint 1: 어떤 데이터들이 필요할지, 그리고 어떤 기능들이 어떤 데이터를 참고하는지, 그리고 데이터들을 각각 어디에 저장할지 고민해보시면 좋습니다.

Hint 2: 문제의 복잡성을 줄이기 위해 피드 업로드 API와 피드 읽기 API만 만든다고 가정합니다.



50분까지 자유롭게 방법을 생각해보세요!

4시 50분까지 진행하고, 쉬는 시간까지 가집니다.