

안녕하세요, 성장하는 개발자 이남곤입니다.



희망 직무 : 백엔드 개발자

Phone : 010-6611-1604

Email : nglee.dev@gmail.com

[GitHub](#) | [블로그](#) | [LinkedIn](#)

늘 성장하고자 합니다.

나에게 부족한 점은 무엇인지를 고민하며 채워가고 있습니다.

의문이 생기는 점이나 모르는 개념은 기록을 하고 공부합니다.

프로젝트 중 새로 알게 된 내용은 세미나 혹은 코드 리뷰 등을 통해 팀원과 공유합니다.

코드 짜는 것을 좋아합니다.

기계적으로 코드를 짜는 것이 아닌 근거와 이유를 가지고 코드를 작성합니다.

중복을 피하고, 재사용성이 높은 코드를 작성하는 것을 좋아합니다.

클린 코드를 지향하고 다른 사람들도 읽기 쉽고 파악하기 쉽게 작성하고자 합니다.

팀원과의 코드 리뷰를 통한 통일된 코드를 지향합니다.

규칙을 준수합니다.

팀 내에서 정한 코드 포맷, 커밋 컨벤션 등을 준수합니다.

규칙을 지키며 일관된 스타일을 유지하도록 합니다.

팀장으로서 이러한 규칙들을 제안하고, 팀원들이 규칙을 지키도록 유도했습니다.

주요 기술



프로젝트에서 가장 먼저 선택하는 언어입니다.
객체지향의 특징, JVM에 대해서 알고 있습니다.



Spring Boot를 활용하여 비즈니스 로직과 REST API를 구현할 수 있습니다.
기본적인 동작 원리를 알고 있습니다.



모든 프로젝트에서 데이터베이스로 MySQL를 사용했습니다.
데이터베이스의 INDEX, 트랜잭션에 대해서 알고 있습니다.



AWS의 EC2, CodeDeploy, S3, CloudWatch, VPC 등등의 서비스를 사용해
본 경험이 있습니다.

교육

삼성 청년 SW 아카데미(SSAFY)

2023.01.03 - 2023.12.29 (수료)

- 그리디 알고리즘, 다이나믹 프로그래밍, 다익스트라와 같은 다양한 문제 풀이 알고리즘들을 익혔습니다.
- 3번의 팀 프로젝트를 진행하며 협업 경험을 쌓았습니다. Spring Boot, MySQL, AWS, Docker 등등의 기술 스택들을 접해볼 수 있었습니다.
- 동료들과 함께 알고리즘 풀이 스터디와 CS 스터디를 진행했습니다.

네이버 커넥트재단, 부스트캠프 웹·모바일 챌린지 과정

2022.07.18 - 2022.08.12 (수료)

- CS 지식들을 직접 구현해보는 미션들이 주어지고, 그에 필요한 CS 지식 등은 직접 찾아보며 구현합니다.
공부한 내용이나 구현 내용을 팀원들과 공유합니다.
- 개발자의 지식 공유 문화를 배우고 지속 가능한 성장을 배웠습니다.

학력

상명대학교(서울), 컴퓨터과학과

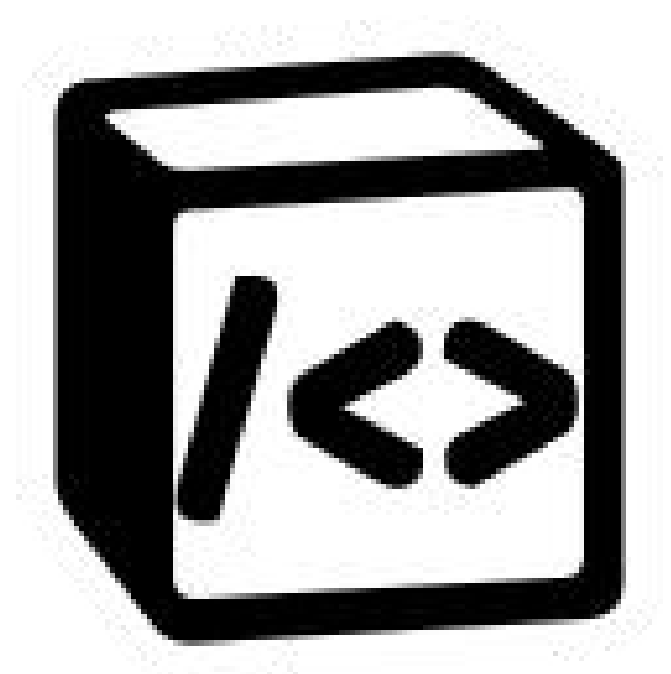
2017.03 - 2023.02 (졸업)

자격증

SQLD(SQL개발자)

2023.07.07 취득

목차



whileTrue

다시 풀고 싶은 알고리즘 문제를 사용자의
노선에 저장하는 크롬 확장 프로그램

개인프로젝트



readed(리디드)

도서의 여러 정보를 한 곳에서 모아볼 수
있는 서비스

팀 프로젝트 (백엔드, 인프라)



opentheDoor

방탈출 카페 정보 제공 및 예약 플랫폼

팀 프로젝트 (백엔드, 인프라)



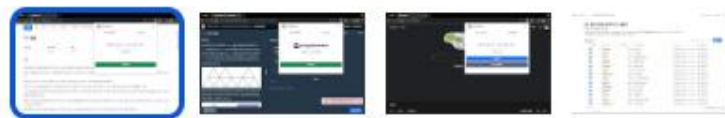
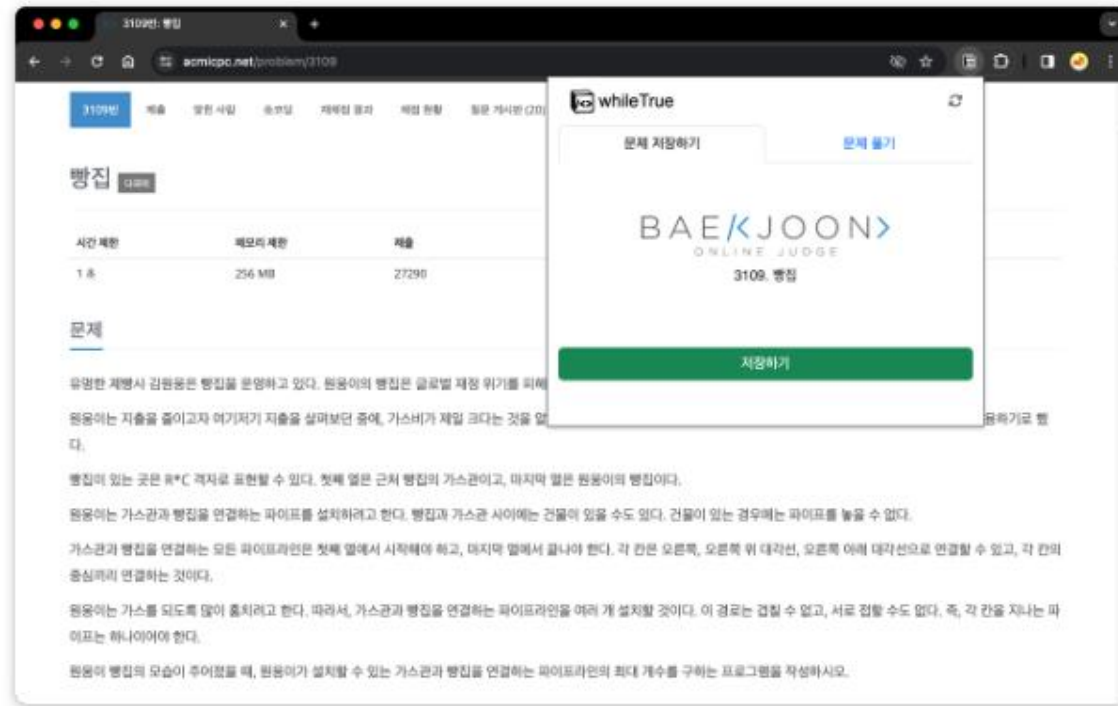
5.0 ★ (평점 2개)

확장 프로그램

교육

5 사용자

Chrome에서 삭제



서비스 소개

[Chrome Web Store](#) | [GitHub](#) | [\(블로그\)프로젝트 회고](#)

다시 풀고 싶은 알고리즘 문제를 사용자의 노선에 저장하는 크롬 확장 프로그램

- 다시 풀고 싶은 알고리즘 문제를 사용자의 노선에 손쉽게 기록할 수 있도록 돕는 크롬 확장 프로그램입니다.
- 저장된 문제들 중에서 임의로 뽑아 다시 풀어볼 수 있습니다.

구성원

개인 프로젝트

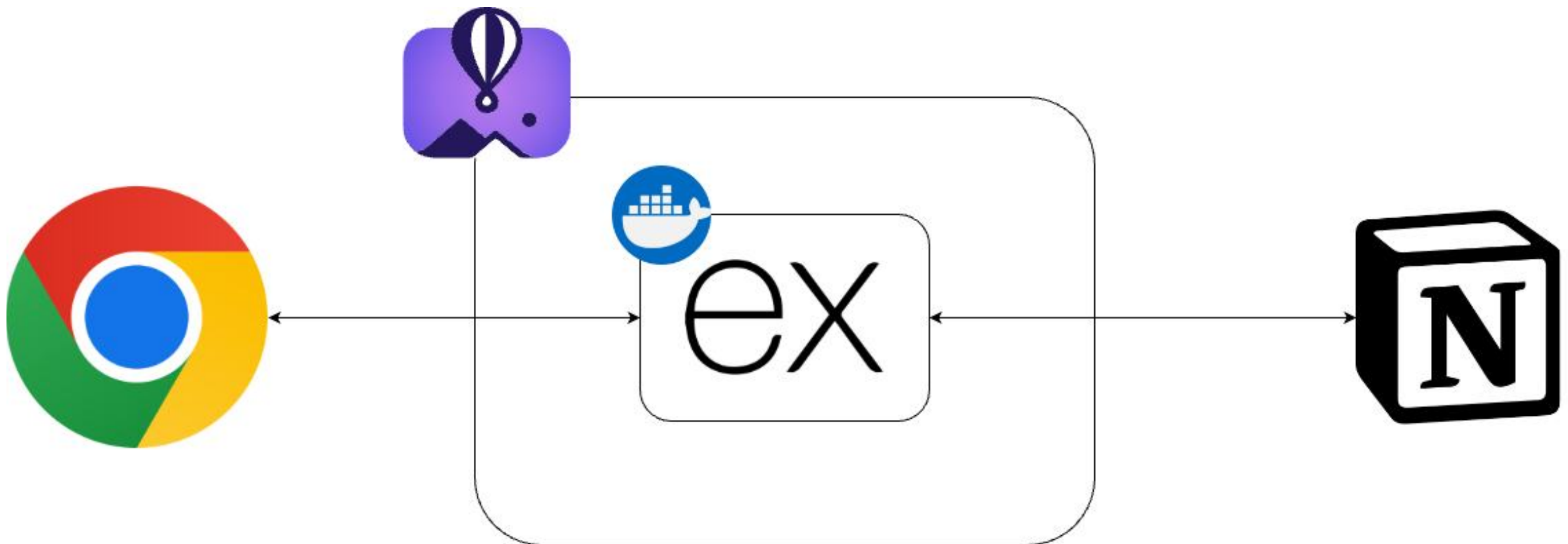
개발 기간

2024.01.16 - 2024.03.04 (현재 배포 중)

기술 스택

- 크롬 확장 프로그램
Typescript, React
- 백엔드
Notion API, Typescript, Express, RDS
- 인프라
Fly.io, GitHub Actions, Docker

시스템 아키텍처



구현 내용

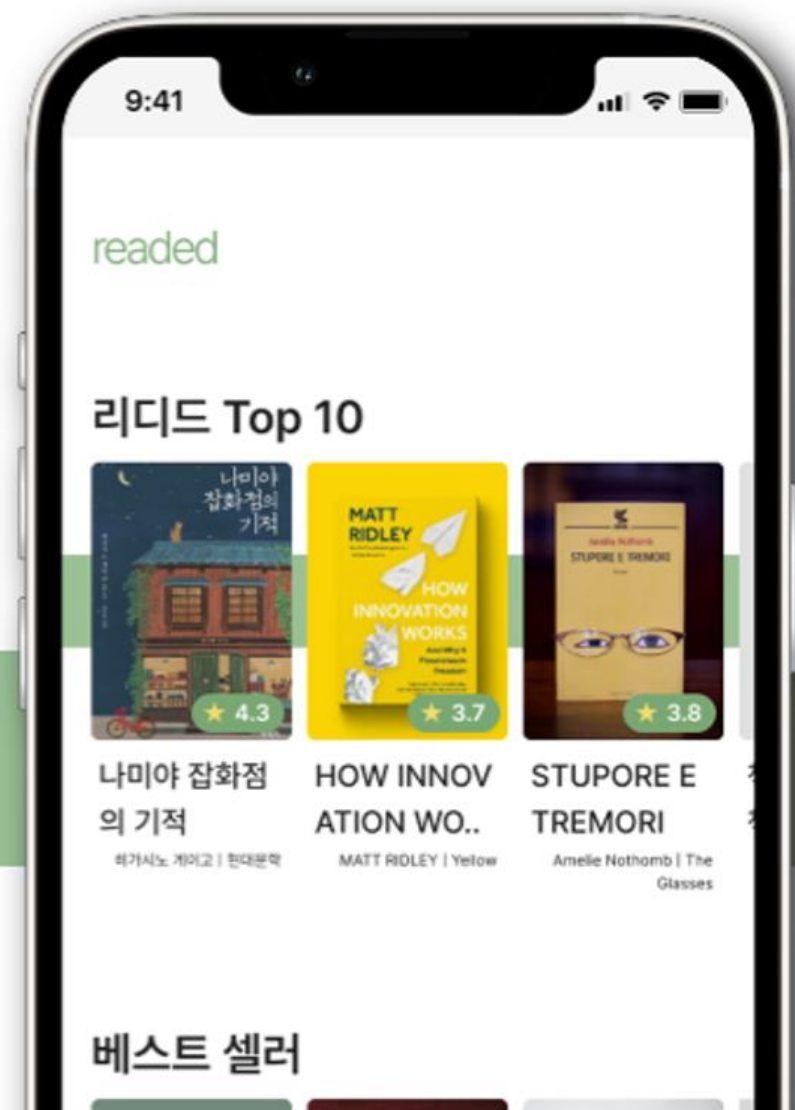
- Notion API를 활용한 사용자 데이터 접근
- Typescript, React를 활용한 크롬 확장 프로그램 구현
- Spring, Express를 활용한 백엔드 구현 → 이후 서버 경량화와 비용 절감을 위해 Express 서버만 유지했습니다.
- EC2를 활용한 인프라 구축과 GitHub Actions, CodeDeploy를 활용한 CI/CD 구축 → 이후 비용 절감을 위해 AWS에서 Fly.io로 옮겼습니다.
- AWS ALB의 HTTP 502 문제 해결

프로젝트 회고

- 크롬 웹 스토어에 출시될 때 뿌듯함을 느낌
- 개발을 시작하기 전 사전 조사 및 구상의 중요성을 느낌
 - 개발을 진행하면서 **Notion API의 CORS 문제와 DTO 유지보수의 어려움**이 있었습니다.
 - 새로운 기술 스택이나, 백엔드와 인프라 도입 등으로 계획보다 개발 기간이 늘어났습니다.
- SSAFY를 통한 성장을 느낌
 - SSAFY에서의 성장 덕분에 큰 어려움 없이 백엔드와 인프라를 빠르게 도입할 수 있었습니다.

readed

독서로 성장하는 나만의 서재



서비스 소개

[GitHub](#)

도서의 이북 서비스로의 링크를 더붙어 여러 정보를 제공하는 서비스

- 도서의 저자, 요약 등의 정보와 여러 이북 서비스 링크를 제공합니다.
- 도서에 대한 별점과 평가를 매길 수 있고, 다른 사람들과 공유할 수 있습니다. 자신만 볼 수 있는 독서 기록을 기록할 수 있습니다.

구성원

- FE - 2명
- BE - 4명 (Infra 2명)
- 담당 역할 - 백엔드 팀장, 백엔드, 인프라

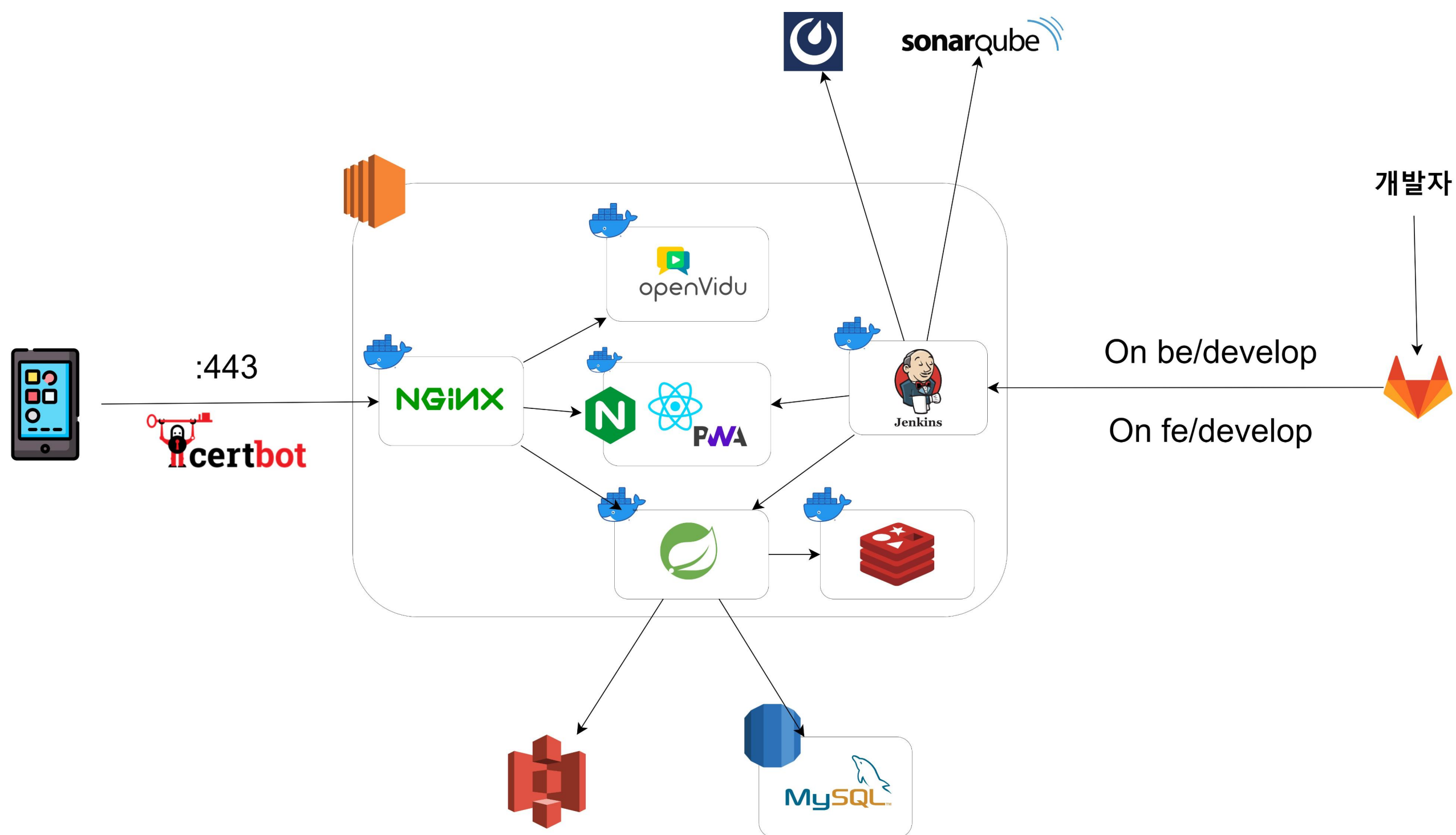
개발 기간

2023.07.04 - 2023. 08.18 (약 6주)

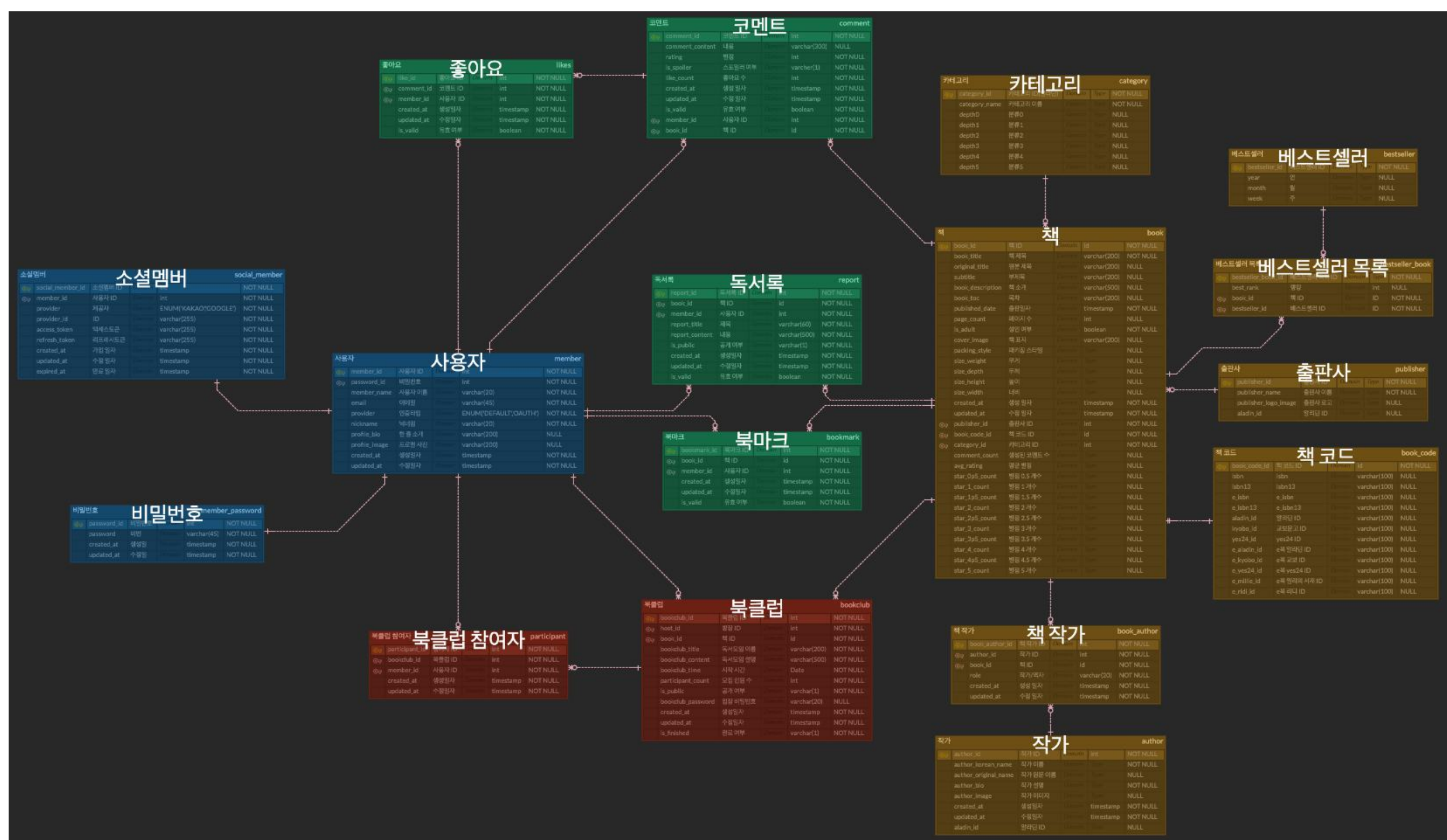
기술 스택

- 백엔드
Java, Spring Boot, Spring Data JPA, Python, Selenium, BeautifulSoup, MySQL
- 인프라
AWS EC2, Jenkins, Docker, Nginx
- 협업
GitLab, Jira, Notion, Mattermost

시스템 아키텍처



ER 다이어그램

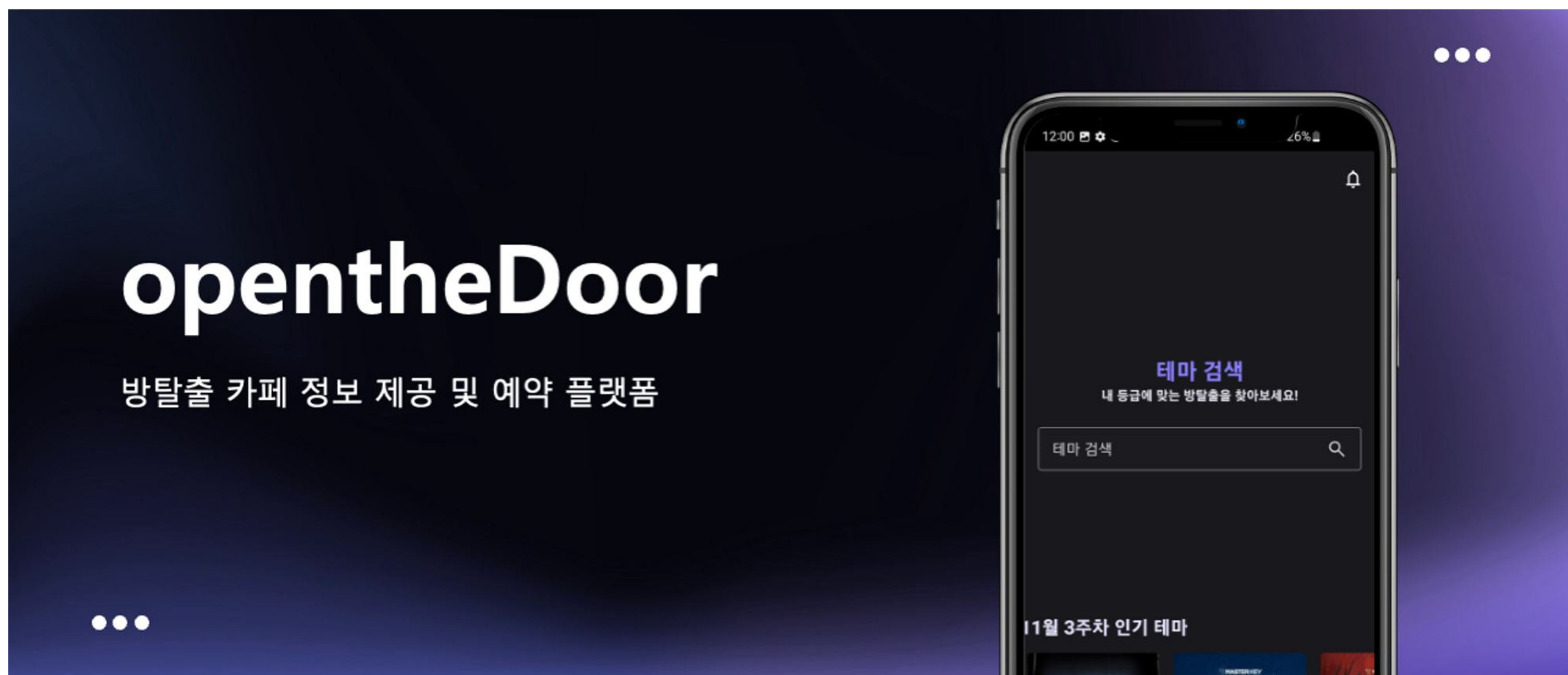


기여한 점

- 도서 정보 크롤링을 통한 DB 구축
 - Python, Selenium을 활용한 크롤링과 OpenAPI를 활용하여 도서 데이터베이스를 구축했습니다.
 - 도서, 저자, 출판사의 연관 관계를 고려하여 크롤링을 했습니다.
- 도서 관련 백엔드 API 개발
- 인프라와 젠킨스 CI/CD 파이프라인 및 Exception 알림 구축
- 코드 리뷰에 적극 참여
 - 팀원들의 Merge Request 시에 코드 리뷰에 적극적으로 참여했습니다.
 - 코드 통일성을 유지하고 조금 더 깔끔한 코드를 작성할 수 있도록 유도했습니다.
 - 팀원의 **JPA N+1 문제를 발견하고 이를 해결**했습니다.
- 프로젝트 후 인프라 및 CI/CD 세미나 진행

프로젝트 회고

- 반 내의 약 60명의 학생 중 우수 학생을 뜻하는 베스트 멤버에 선정되었습니다.
- 첫 팀 협업 프로젝트
 - 처음으로 진행해본 협업 프로젝트입니다.
 - 팀 개발의 진행 방식과 소통 방식을 배웠습니다.
- 첫 인프라와 CI/CD 구축
 - 인프라와 CI/CD에서 주로 사용되는 도구들을 경험해보았고, 전체적인 흐름을 배울 수 있었습니다.
 - 처음으로 CI/CD를 경험해보았습니다. 프로젝트가 진행되면서 CI/CD의 중요성을 몸소 깨달았습니다.
- 무중단 배포의 아쉬움



서비스 소개

[통합 GitHub](#) | [백엔드 GitHub](#)

방탈출 카페 정보와 예약 서비스까지 제공하는 서비스

방탈출 예약을 위해서는 각각의 방탈출 카페 플랫폼에 접근해야 했던 기존의 방식에 벗어나 “오픈더도어” 서비스에서는 예약 기능까지 지원하여, 방탈출 카페와 관련된 모든 과정을 한 곳에서 진행할 수 있습니다.

구성원

- FE - 2명
- BE - 4명 (Infra 1명)
- 담당 역할 - 팀장, 백엔드, 인프라

개발 기간

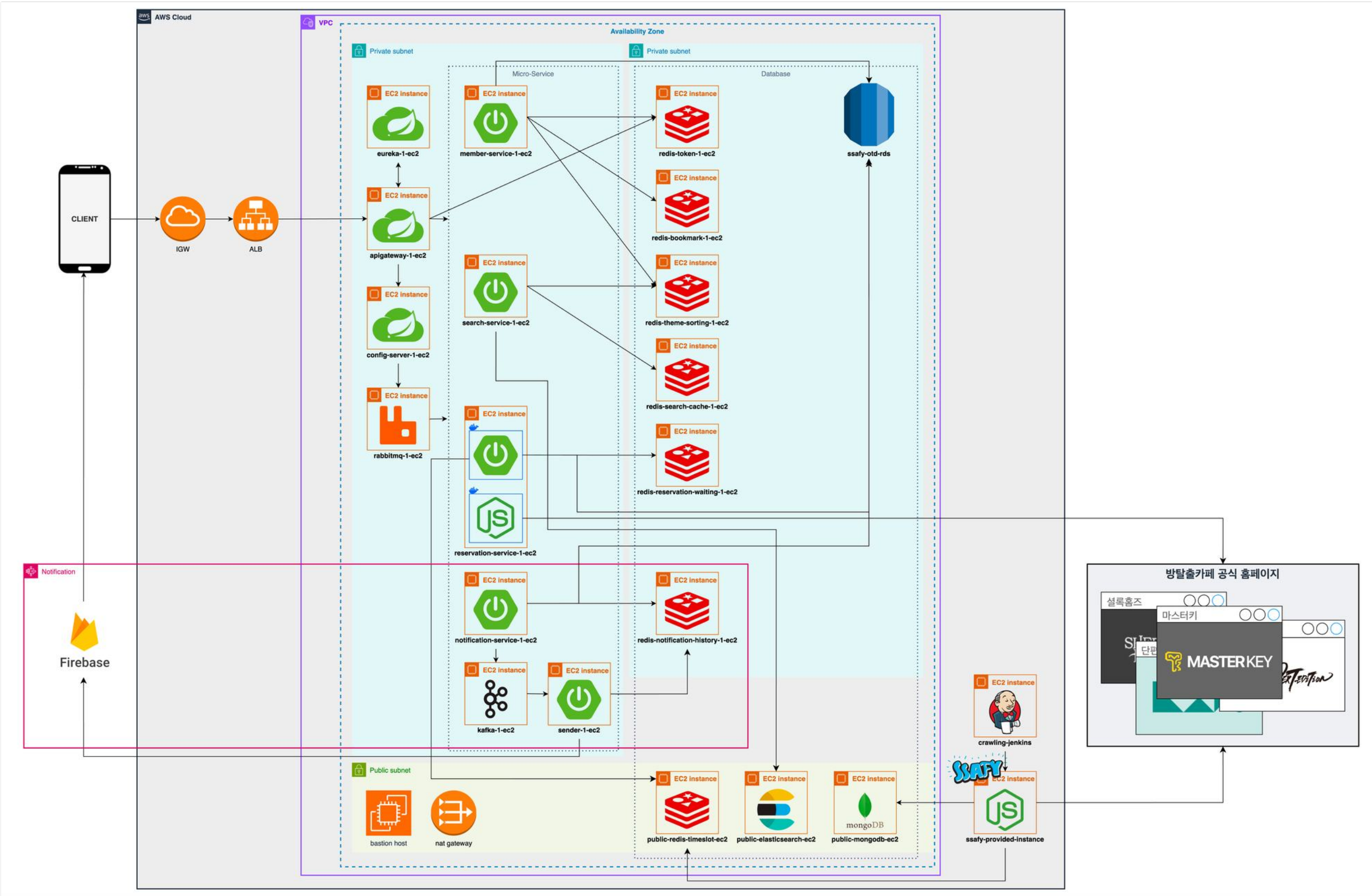
2023.10.09 - 2023.11.17 (약 6주)

기술 스택

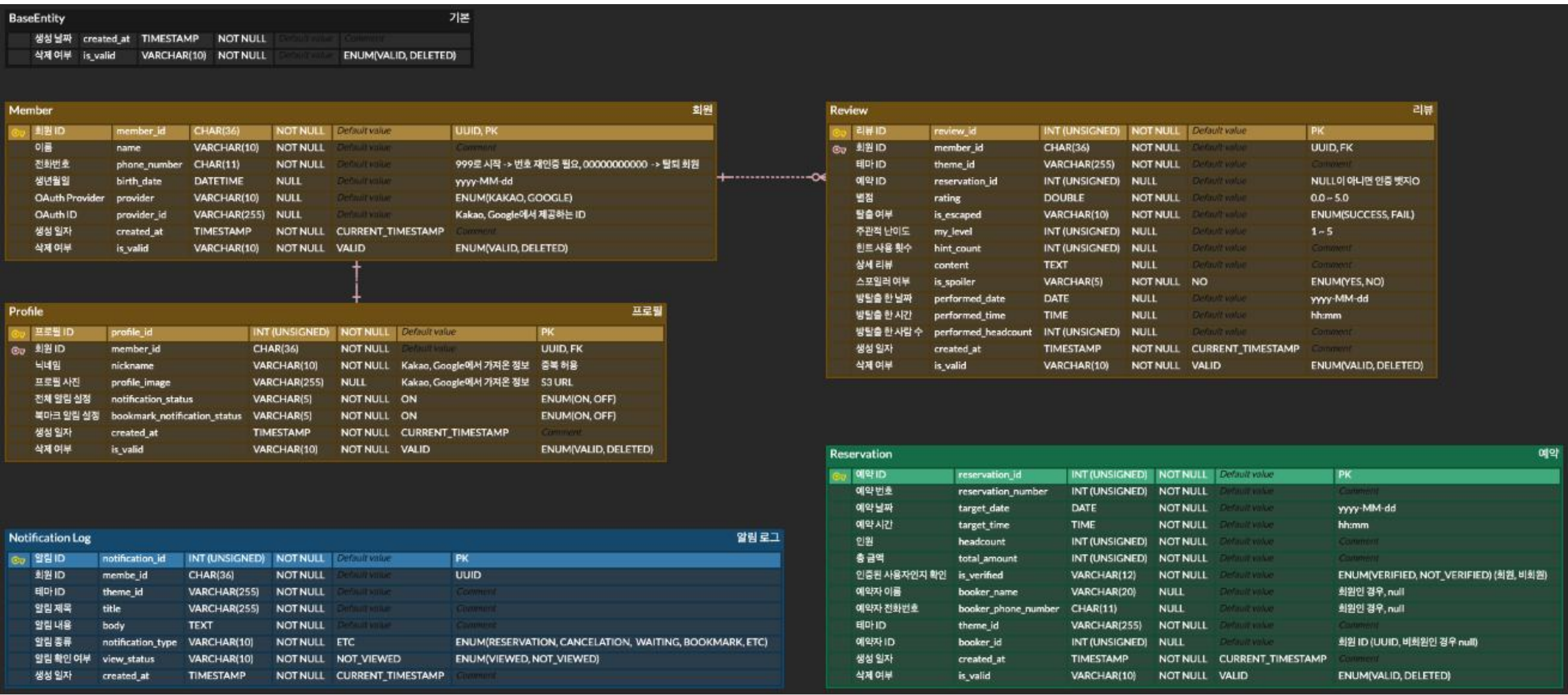
- 백엔드
MSA, Java, Spring Boot, Spring Data JPA, Node.js, Express, puppeteer, Cheerio, MySQL
- 인프라
GitHub Actions, Jenkins, AWS EC2, AWS CodeDeploy, AWS S3, AWS CloudWatch, Docker
- 협업
GitHub, Notion, Mattermost



시스템 아키텍처



ER 다이어그램





기여한 점

- MSA에 기반한 인프라 구축
 - VPC를 구축했고, 마이크로 서비스 4개와 API Gateway, Eureka Server를 포함하여 **총 22개의 EC2 인스턴스를 관리**했습니다.
- 백엔드 예약 서비스 구축
- 방탈출 테마 시간 정보 크롤링
 - 여러 방탈출 테마들의 시간 정보를 크롤링하는 서버를 구축했습니다.
 - 사용자는 이 정보를 통해 테마의 현재 비어있는 시간대를 파악할 수 있습니다.
- 예약 동작 기능 구현
 - “오픈더도어” 어플에서 사용자가 예약을 하면, 실제 방탈출 예약 사이트에 접근하여 실제 예약 기능을 수행합니다.

프로젝트 회고

- 반 내의 10팀 중에서 2위에 해당하는 우수 프로젝트로 선정되었습니다.
- 첫 MSA 도전
- VPC를 활용한 MSA 인프라 구축
 - 총 22개의 EC2 인스턴스를 관리했습니다.
 - 많은 수의 인스턴스를 관리할 때는 모니터링 툴이 필수적이라는 것을 느꼈습니다.
 - 시간 상, 비용 상 하나의 AZ밖에 사용하지 않았습니다. 두 개 이상의 AZ를 활용한 로드밸런싱을 경험해보지 못해 아쉬움이 남습니다.
- 장애 확산 방지의 필요성과 MSA 도입 필요성 재고
 - 마이크로 서비스간 결합도가 높고 장애 확산 방지가 되어 있지 않아 MSA의 장점을 온전히 활용하지 못했습니다.
 - 개발, 관리 복잡성과 비용 증가로 신중히 MSA를 도입해야 한다는 것을 배웠습니다.

감사합니다
이남곤입니다