

배준일

Backend Engineer

Email	bjo6300@naver.com
Mobile	010-6216-2388
Github	https://github.com/bjo6300
Blog	https://velog.io/@bjo6300

Summary

일상생활에서 마주하는 불편함을 인지하고 적절한 기술을 활용하여 해결하는 것이, 제가 추구하는 개발자의 모습이자 제가 수행할 역할이라고 생각합니다. 네이버 카페에서 유기 동물 봉사 신청 과정에서 불편함을 느껴 동물 보호소와 봉사자를 매칭하는 서비스를 개발한 경험이 있습니다.

Tech Skills

- **Language** : Java
- **Backend** : Spring boot (Data JPA, QueryDsl)
- **DevOps** : Docker, Github Actions
- **DataBase** : MySQL, Redis
- **Testing** : JUnit, Mockito
- **Cloud** : AWS (EC2, S3, RDS)

Experience

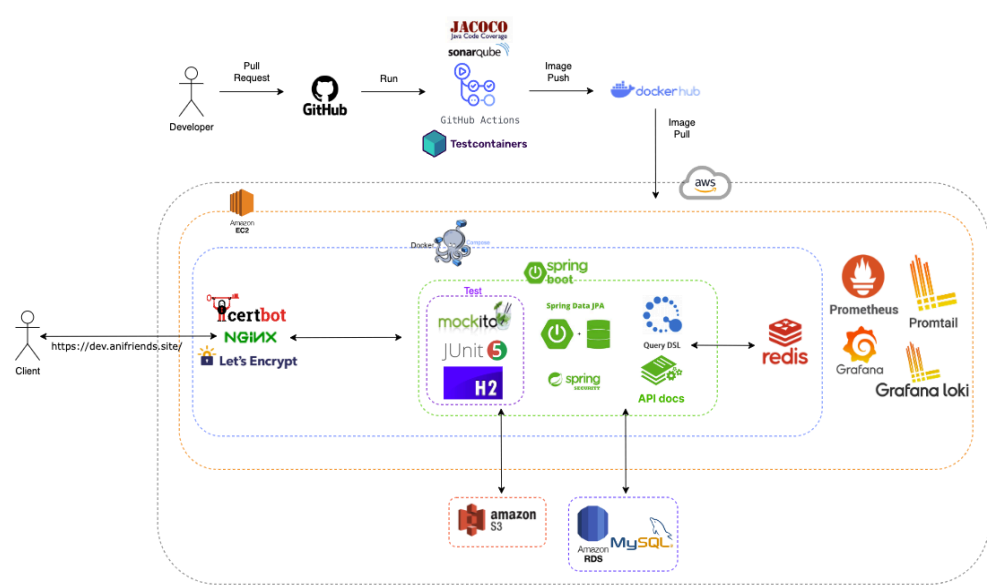
2022.08 ~ 현재	Techeer : SW 개발자 커리어 그룹 - 팀 프로젝트, 해커톤, 커피챗, 스터디 등 다양한 개발 활동 참여
2023.04 ~ 2023.05	Google Cloud Study Jam [링크]
2022.11 ~ 2023.01	한국공학대학교 연구실 학생연구원 - springboot 프로젝트 리팩토링
2022.12.06	2022 한이음 ICT멘토링 공모전 - 입선
2022.10.12	한국공학대학교 컴퓨터공학부 종합설계 성과확산 프로그램 - 동상
2022.08 ~ 2022.10	성남시 AI 프리인턴십 [링크]

Education

2023.06 ~ 2023.12	프로그래머스 웹 백엔드 데브코스 - 코드 리뷰 기반의 미션 수행 / 팀 프로젝트를 통한 협업 경험 [동료평가]
2017.03 ~ 2023.02	한국공학대학교 컴퓨터공학과

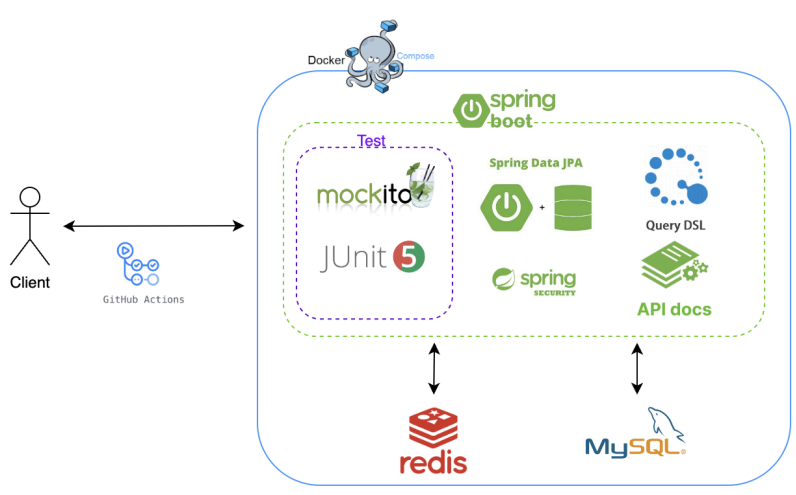
Project

Anifriends

프로젝트 소개	보호소와 봉사자의 소통 프로세스 개선하고, 봉사자의 신원과 신청현황 등에서 발생하던 불편함을 개선한 서비스
	기간 : 2023.10 ~ 2023.12 팀 구성 : 프론트엔드 4명, 백엔드 4명
기술 스택	Java 17, Spring boot, Spring Data JPA, Querydsl, MySQL, H2, Redis Docker, GitHub Actions, Nginx, AWS(EC2, S3, RDS, ElastiCache) Prometheus, Grafana, Loki, Promtail
링크	Github : https://github.com/Anifriends/Anifriends-Backend
개발 내용	 <p>봉사 모집 글 목록 조회 성능 향상 (419ms → 105ms)</p> <ul style="list-style-type: none">- Querydsl을 이용해 동적 쿼리 구현 및 무한 스크롤에 적합한 커서 기반 no-offset 페이지네이션 구현- Redis에 봉사 모집 글 개수 캐싱 <p>보호 동물 목록 조회 성능 향상 (1067ms → 259ms) [링크]</p> <ul style="list-style-type: none">- QA 중 목록 조회 API 호출 시 1000ms 이상의 시간이 소요됨을 확인- 정렬 조건에 해당하는 생성 시간을 인덱스로 설정하여 조회 성능 개선- ToMany 관계의 페이징을 BatchSize에서 DTO Projection으로 개선- 21개의 미사용 컬럼 최적화 <p>인원 마감에서 발생할 수 있는 동시성 문제를 비관적 락으로 해결 [링크]</p> <p>전체 테스트 코드 중 ⅓ (약 220개)를 작성. 테스트 커버리지 80% 이상 유지</p> <ul style="list-style-type: none">- 통합테스트를 위한 추상클래스를 만들어 ApplicationContext를 캐싱해 테스트 수행 시간 단축- ExecutorService, CountDownLatch를 이용해 동시성 검증 통합 테스트 코드 작성- 독립적인 Redis 테스트 환경을 구축하기 위해 TestContainer 적용 <p>존재하지 않는 객체와 협력</p> <ul style="list-style-type: none">- S3가 연동되지 않은 상황에서 런타임에 다른 서비스에 사용하기 위한 객체 필요하여 인터페이스를 이용해 해결- 기존 스프린트에 영향을 주지 않고 SOLID 원칙을 준수하여 개발

<p>빌드 및 배포 시간 단축 [링크]</p> <ul style="list-style-type: none"> - (기존) EC2에 jar 파일을 수동 배포 - (변경) Docker, Github Actions를 이용해 CI/CD 파이프라인을 구축하여 빌드 및 배포 자동화 - 팀원의 개발 생산성 향상 <p>로그 파일 크기 증가로 인해 EC2 디스크 용량이 가득 차 서버 실행이 불가능한 문제 발생</p> <ul style="list-style-type: none"> - EC2 디스크 용량이 적어지면 S3에 압축해서 백업 후 서버 로그를 초기화하는 script로 해결 [링크]
--

NaBmart

프로젝트 소개	배달의 민족 B마트처럼 상품을 주문하고 배달하는 서비스
	기간 : 2023.08 ~ 2023.09 팀 구성 : 백엔드 6명
기술 스택	Java 17, Spring boot, Spring Data JPA MySQL, H2, Redis Docker, GitHub Actions
링크	Github : https://github.com/prgrms-be-devcourse/BE-04-NaBmart
개발내용	<div style="text-align: center;">  <p>The diagram illustrates the system architecture. On the left, a 'Client' (represented by a stick figure) interacts with 'GitHub Actions' (represented by a blue octopus icon). A double-headed arrow connects the Client to a large blue rounded rectangle representing the application container. Inside this container, at the top, are 'Docker' and 'Compose' icons. Below them is a green 'spring boot' logo. A dashed green box labeled 'Test' contains 'mockito', 'JUnit 5', 'Spring Data JPA', 'spring security', 'Query DSL', and 'API docs'. Below the 'Test' box, there are two database icons: 'redis' (red cube) and 'MySQL' (blue fish). Double-headed arrows connect the application container to both the Redis and MySQL databases.</p> </div> <p>상품 목록 조회 성능 향상 (160ms → 12ms) [링크]</p> <ul style="list-style-type: none"> - Redis에 평균 평점, 리뷰 수 캐싱 - Look aside, Write through 캐시 전략을 이용해 데이터의 일관성 유지 - 데이터 정합성을 위해 Redis의 데이터를 동기화하는 스케줄러 구현 <p>라이더가 동시 배차 요청 상황에서 발생할 수 있는 동시성 문제를 낙관적 락으로 해결 [링크]</p> <p>전체 테스트 코드 중 ⅓ (약 120개)를 작성. 테스트 커버리지 80%이상 유지</p> <ul style="list-style-type: none"> - 독립적인 테스트 환경을 구축하기 위해 TestContainer 사용 [링크] - 조회 테스트용 Fixture 환경을 재사용하여 효율적인 테스트 코드 작성