

유성호 이력서

Java / Spring Backend Developer

Java/Spring 기반으로 조회·검색 API, 커뮤니티, 스토어/결제, AI 기능 연동을 구현하며 성능 개선과 데이터 정합성 문제를 함께 다뤘은 백엔드 개발자입니다.

SQL 실행계획을 바탕으로 500만 건 조회 병목을 14.2초에서 60ms까지 개선했고, Redis 캐싱, Elasticsearch 검색 분리, Kafka 이벤트 처리, Kubernetes 운영 전환까지 경험했습니다.

Contact

- Email: shyu6370@gmail.com | Phone: 010-8250-6370
- GitHub: <https://github.com/Dorosiya>
- Service: <https://www.pawbridge.kr>

Skills

Backend

- Java 17/21 · Kotlin · Spring Boot 3 · Spring Cloud · Spring Data JPA · Spring Security · QueryDSL · Spring Batch · Python · FastAPI

DB / Cache / Messaging

- MySQL · PostgreSQL · Redis · Elasticsearch · Kafka · Debezium CDC

Infra / DevOps

- AWS EC2 · Docker · GitHub Actions · Kubernetes · Helm · Vagrant

Monitoring

- Prometheus · Grafana · Zipkin

AI / LLM

- OpenAI API · Gemini API · PyTorch · DINOv2

Frontend / View

- React · TypeScript · Vue.js · JSP · Thymeleaf · JavaScript · jQuery

Tools

- Git · IntelliJ · Postman

AI Development

- Codex · Claude Code · Cursor · Antigravity · CodeRabbit

Projects

1) PawBridge | Backend | 2025.11 – 현재 | 운영 중 | 2인

공공데이터 기반 유기동물 조회, 검색, 커뮤니티, 스토어/결제 기능을 제공하는 입양 플랫폼입니다. Java/Spring 기반 API 개발과 Python/FastAPI AI 기능 분리, 검색 성능 개선, 캐싱, 이벤트 정합성, Kubernetes 운영 전환까지 담당했습니다.

운영 URL

pawbridge.kr

주요 기술 스택

Java 17 · Spring Boot 3.4 · Python · FastAPI · MySQL · Redis · Kafka · Elasticsearch · Kubernetes · Helm · Docker · GitHub Actions · Prometheus · Grafana · Zipkin · Gemini API · React · TypeScript

- 공공데이터 동기화와 검색 색인 후속 처리가 비즈니스 트랜잭션과 분리될 경우 누락 가능성이 있다고 판단해, 비즈니스 데이터와 Outbox 이벤트를 같은 트랜잭션에 저장하고 Debezium CDC로 Kafka에 전달하는 구조를 적용했습니다.

- 동물 목록·스토어 상품 검색에서 LIKE 기반 검색의 한계를 줄이기 위해 Elasticsearch를 도입하고, CUD는 MySQL, 키워드 검색 조회는 Elasticsearch로 분리했습니다. Analyzer 기반 토큰화와 BM25 관련도 점수를 활용해 키워드 검색 품질을 높이고, 검색 기능을 독립적으로 확장할 수 있는 구조를 마련했습니다.
- Python · FastAPI 기반 AI 기능을 별도 서비스로 분리해 DINOv2 이미지 유사도 추천을 구현하고, 보호동물 상세 페이지에는 Gemini 기반 입양 준비 도우미 챗봇을 추가해 익명 세션 저장, 내부 API 인증, Redis 기반 호출 제한을 적용했습니다.
- 이벤트 처리 순서가 중요한 증감형 집계 구간에서 Kafka 단일 토픽과 키 기반 파티셔닝을 적용해 동일 개체 기준 이벤트가 같은 파티션에서 순서대로 처리되도록 구성했습니다.
- 캐시 만료 시 여러 파드가 동시에 DB를 조회·갱신하는 Cache Stampede를 줄이기 위해 Redis 분산 락으로 갱신 주체를 제한하고, Double-Check로 대기 요청이 갱신된 캐시를 재사용하도록 했으며, TTL Jitter로 만료 시점을 분산해 DB 부하와 불필요한 중복 요청을 줄였습니다.
- EC2 5-Node 운영 종료 후에도 Kubernetes 운영 구조를 검증하기 위해 로컬 VM 3-Node 클러스터로 전환하고, Helm · ConfigMap · Secret 기반 배포 구성을 직접 검증했습니다.
- Prometheus · Grafana · Zipkin 기반 관측 환경과 k6 부하 검증, Slack 알림 연동으로 운영 지표를 모니터링하고 이상 징후를 빠르게 확인했습니다.

2) 셋별 | Backend | 2025.08–2025.09 | 4인

AI 모의면접과 개발자 커뮤니티를 결합한 커리어 플랫폼으로, 대용량 조회 최적화와 외부 AI API 장애 대응을 설계한 프로젝트입니다.

주요 기술 스택

Java 17 · Spring Boot 3 · JPA · MySQL · Redis · OpenAI/Gemini API · AWS EC2 · Docker · GitHub Actions

- 게시글 목록 조회에서 1:N JOIN과 DISTINCT로 Row Explosion이 발생해 평균 14.2초 지연이 발생했고, 스칼라 서브쿼리 전환과 복합 인덱스 재설계로 500만 건 조회 성능을 60ms까지 줄였습니다.
- 게시글 목록·검색·상세 조회에 Redis Cache-Aside를 적용하고 캐시 무효화 범위를 나눠 반복 조회 구간 응답 시간을 60ms에서 15ms까지 줄였습니다.
- 실시간 면접 흐름에서 매 턴마다 수행하던 상세 점수 계산을 종료 후 배치 평가로 전환하고, 기능별로 모델을 분리 적용해 LLM 호출 수를 21회에서 11회로 줄였으며 nextTurn 평균 응답 시간을 8.5초에서 2.3초로 개선했습니다.
- 외부 LLM API 호출 구간에 Timeout · Retry · Circuit Breaker를 적용하고, 기능별 실패 처리 흐름을 분리해 AI API 장애가 서비스 전체로 전파되지 않도록 했습니다.
- 게시글·댓글 등록 후 실행되는 AI 기반 콘텐츠 검수 로직을 Spring Event로 분리하고, @TransactionalEventListener(AFTER_COMMIT)을 적용해 트랜잭션 완료 후 후속 로직이 실행되도록 조정했습니다.

3) OTT-Moa | Backend | 2025.06–2025.07 | 4인

실시간 채팅 기반 OTT 파티 매칭 서비스로, 다중 서버 메시지 동기화와 선착순 동시성 제어를 구현한 프로젝트입니다.

주요 기술 스택

Java 17 · Spring Boot 3 · PostgreSQL · Redis Pub/Sub · WebSocket/STOMP · AWS EC2 · GitHub Actions · Vue.js

- 단일 서버 메모리에 종속된 WebSocket 구조를 개선하기 위해 Redis Pub/Sub을 도입해 다중 서버 환경 메시지 동기화가 가능하도록 구성했습니다.
- 이 프로젝트에서는 메시지 영속성보다 실시간 fan-out과 운영 단순성이 더 중요하다고 판단해 Kafka 대신 Redis Pub/Sub을 선택했습니다.
- 선착순 가입 구간에서 정원 초과가 실제 환불 이슈로 이어질 수 있어, 재시도 비용보다 무결성을 우선해 비관적 락으로 동시성 충돌 가능성을 줄였습니다.
- Fetch Join과 batch fetch 설정을 적용해 N+1 문제를 줄이고 조회 쿼리를 28회에서 2회로 줄였습니다.

4) Ops Console | Backend | 2026.06 – 현재 | ICT콤플렉스 기업 협업 실무 프로젝트

기업 내무 운영을 위한 콘텐츠 관리 백오피스를 신규 구축하고 있습니다.

Kotlin/Spring 기반 기능 구현과 기업 측 코드 리뷰·테스트 반영을 진행하고 있습니다.

Awards & Certificates

- 정보보안기사(필기합격, 2026)
- IITP 우수성과 공유컨퍼런스 우수상 (2025)
- 정보처리기사 (2024)
- SQLD (2024)

Education

삼육대학교
경영학과
2015.03 – 2021.02 졸업

Training

- 한국소프트웨어산업협회 | MSA 기반 Full Stack 개발 전문가 양성과정 수료 | 2025.03–2025.09