



# 김윤기

## Backend Developer



- 010-3385-2507
- kalsagansteve@gmail.com
- github.com/kalsteve
- velog.io/@kalsteve98

## Summary

소규모 팀 프로젝트에서 FastAPI·MySQL·Redis로 REST & WebSocket API를 개발하며 최적화한 경험이 있습니다. GitHub Actions + AWS CodeDeploy를 활용해 Blue/Green 무중단 배포 파이프라인을 직접 구축하며 배포 자동화 기초를 다졌습니다. BLE·Android 경험으로 디바이스와 클라우드 백엔드를 엔드투엔드로 통합할 수 있어 IoT·모빌리티 제품을 빠르게 검증할 수 있습니다. 코드 리뷰·문서화·에러 로그 분석에 꾸준히 참여하며 협업과 지속 학습을 중시합니다.

## Project

### Brain Washer (2024.06 - 2024.08) - [github](#)

실존하는 특정 인물이 독설로 동기 부여해주는 서비스

FastAPI, MySQL, AWS(CloudFront, EC2, RDS, CodeDeploy), GitHub Actions, Docker, Redis, Nginx

#### 비동기 I/O 파이프라인 재설계 - (Blog)

- **문제**: HTTP Long Polling과 동기식 TTS 렌더링으로 인해 최대 180s의 TTFB가 발생하였습니다.
- **시도**: FastAPI async 엔드포인트와 asyncio.Queue 프로듀서-컨슈머 패턴을 적용하고, SSE 스트리밍으로 연결을 업그레이드하였습니다.
- **결과**: p95 TTFB를 **180s → 12s(-93 %)**로 단축하고 Core Web Vitals INP를 **200ms** 미만으로 유지하였습니다.

#### Redis Look-Aside 캐시로 QPS 스파이크 흡수

- **문제**: 문제 피크 타임 DB QPS가 3k/s를 초과하며 RDS CPU 사용률이 90 %에 도달하였습니다.
- **시도**: Redis LRU+TTL 캐시를 Look-Aside + Write-Back 패턴으로 도입하였습니다.
- **결과**: RDS QPS를 40 % 절감하고 p95 **응답 지연을 추가로 35 % 단축**하였습니다.

#### Blue/Green 무중단 배포 자동화

- **문제**: 제 배포 도중 ALB 대상 인스턴스가 drain 상태로 전환되지 않아 간헐적 502 에러가 발생하였습니 다.
- **시도**: GitHub Actions에서 **CodeDeploy Blue-Green 전략**을 적용하고 Target Group Health Threshold를 3/5로 조정하였습니다.
- **결과**: 서버 중단 시에도 서비스가 중단되지 않았으며, 알림을 통해 바로 문제를 해결 할 수 있었습니다.

#### 멀티-스테이지 CI/CD 고도화

- **문제**: 단일 서버 구성으로 프론트엔드·백엔드 커플링이 높아 장애 반경이 확대되었습니 다.
- **시도**: GitHub Actions + CodeDeploy빌드로 FE(S3·CloudFront) / BE(Docker·ECR·ECS) **병렬 배포**와 Secret Manager **암호화**를 적용하였습니다.
- **결과**: 배포 시간을 **15분 → 6분**으로 단축하고 컴포넌트 격리로 장애 영향 범위를 최소화하였습니 다.

### Tairot (2023.12 - 2024.02) - [github](#)

인격에 따라 스타일이 다른 AI 타로 마스터가 각자의 방식으로 타로를 풀이해주는 서비스

Node.js, Express, Socket.IO, MySQL, AWS(EC2, RDS, Secrets Manager), Jenkins, Docker, Nginx

#### Nginx rate-limiting & Cloudflare로 L7 DDoS 방어

- **문제**: 50k RPS 규모의 HTTP flood로 EC2 인스턴스가 OOM으로 다운되었습니 다.
- **시도**: **Nginx limit\_req\_zone(150 r/s)** 및 **Cloudflare Managed Rule Set**을 적용하고 Fail2Ban으로 반복 IP를 차단하였습니 다.
- **결과**: 100k RPS까지 서비스를 안정적으로 유지하며 요청 성공률 99.98 %를 달성하였습니 다.

#### Prometheus + Grafana Observability

- **문제**: 메모리 누수와 event-loop block 장애에 대한 근본 원인 분석이 지연되었습니 다.
- **시도**: Node Exporter, MySQL Exporter, Loki 로그 수집을 통해 Grafana 대시보드 6종을 구축하고 Alertmanager를 Slack과 연동하였습니 다.
- **결과**: 장애 탐지 시간을 **30분 → 5분으로 축소**하고 SLO 99.9 %를 달성하였습니 다.

### IT -infra (2024.09 - 2025.06) - [github](#)

골목길 보행자와 자동차와의 사고를 비콘을 통해 방지하는 서비스

Android, Jectpack compose, FastAPI, MySQL

#### Foreground Service로 BLE 스캔 지속성 확보

- **문제**: Android Doze/App Standby 정책으로 백그라운드 BLE 스캔이 중단되었습니 다.
- **시도**: Foreground Service + Notification (Ongoing)를 유지하였습니 다
- **결과**: 스캔 성공률을 **62 % → 98 %**로 향상하고 배터리 소모를 **4.2 %/h → 2.8 %/h**로 절감하였습니 다

#### Compose StateFlow 기반 실시간 UI 리팩터링

- **문제**: Intent 브로드캐스트 방식에서 UI 스레드 오버로드로 dropped frame이 발생하였습니 다.
- **시도**: **ViewModel + StateFlow** 패턴으로 비콘 스트림을 바인딩하고 Recomposition Scope를 최적화하였습니 다.
- **결과**: 경고 latency를 1s → 320ms로 단축하고 Drop Frame 비율을 14 % → 3 %로 감소하였습니 다.

#### Kalman + EMA 하이브리드 필터로 RSSI 정밀도 향상

- **문제**: RSSI  $\pm 10$  dBm 노이즈로 거리 추정 오차가 발생하였습니다.
- **시도**: 1차 Kalman 필터 후 N=4 지수이동평균을 적용하여 스무딩하였습니다.
- **결과**: 결과 위치 오차를  **$\pm 1.2$  m →  $\pm 0.8$  m(-35 %)로 감소**하였습니다.

# Technical Skills

---

- C 언어 BSD 소켓 API와 pthread 기반 멀티스레딩을 이용해 수백 명이 동시에 접속하는 할리갈리 게임 서버를 구현한 경험이 있습니다. 구조체 기반 패킷 프로토콜, mutex·cond 동기화로 안정적인 에러 복구 루틴을 직접 설계·최적화하였습니다.
- CI/CD 영역에서는 GitHub Actions → AWS CodeDeploy → Docker Compose를 파이프라인으로 연동하고 ALB Blue/Green 전략을 적용하여 무중단 배포(Zero-Downtime)를 달성하였습니다. 배포 전후 헬스 체크·Slack 알림·자동 롤백을 탑재해 평균 배포 시간을 10 분 → 4 분으로 단축하고 실패율을 0%로 유지한 경험이 있습니다.
- Node.js(Express) 마이크로서비스를 구축하면서 OpenAI Chat/Speech API의 단일 응답 한계를 해결하기 위해 WebSocket 파이프라인을 설계, JSON chunk 단위로 음성 스트림을 실시간 전송하여 p95 응답 지연 < 3 초를 달성했습니다.
- 모바일 분야에서는 Android Jetpack Compose와 BLE Scanner API를 활용해 iBeacon(UUID·Major·Minor·RSSI) 기반 충돌 예방 앱을 개발했습니다. Foreground Service으로 Doze/App Standby 환경에서도 스캔 지속성을 확보하고, 권한 흐름·배터리 소비 지표를 직접 튜닝하여 스캔 성공률 62 % → 98 %, 소모 4.2 %/h → 2.8 %/h까지 최적화한 경험이 있습니다.

# Activity

---

Techeer 8기 (2024.08 - )

2024 하계 테커 실리콘밸리 소프트웨어 부트캠프 (2023.06.24 - 2024.08.03)

Techeer partners (2024.03.23 - 2024.06.23)

- 실리콘밸리 엔지니어의 SW개발자 스터디 그룹
- 기술 세션, 프론트엔드 React 프레임워크등의 스터디 활동

2023 동계 테커 실리콘밸리 소프트웨어 부트캠프 (2023.12.26 - 2024.02.03)

Techeer partners (2023.09.12 - 2023.12.23)

- 실리콘밸리 엔지니어의 SW개발자 스터디 그룹
- 기술 세션, 백엔드 Spring Boot 프레임워크등의 스터디 활동

# Education

---

- 한국공학대학교 (2023.3 - 2025.8 졸업예정)
- 임베디드시스템전공 (3.53 - 4.5)