

길현준

Backend Engineer / Platform Part Leader

kboxstar@gmail.com 서울, 대한민국

github.com/kilhyeonjun www.linkedin.com/in/kilhyeonjun 블로그: blog.kilpenguin.com

소개

6개 라이브 게임 환경(12 브랜치 운영)의 공통 백엔드 플랫폼을 설계·운영하고, 신설 Platform Part 리더로서 실시간 게임 채팅·마케팅 데이터 파이프라인·공통 패키지 체계를 주도하는 백엔드 엔지니어입니다. 게임듀오에서 BigQuery Storage Read API 전환으로 TB당 조회 비용 82% 절감, 서버리스 이관으로 처리 범위 60 → 360일 확장, 6,000 동시접속 목표의 실시간 채팅 서버 용량 검증을 수행하고 [Games on AWS 2025](#) 고객 세션에서 사례를 발표했습니다. 4년 11개월 동안 교육·헬스케어·임상시험·게임 4개 도메인에서 DB 최적화, 동시성 제어, 레거시 현대화, 이벤트 기반 아키텍처를 일관되게 수행했습니다. 최근에는 AI 도구를 단순 코드 생성기가 아니라 TDD·통합 테스트·리뷰·근거 검색으로 산출물을 검증하는 개발 품질 체계에 통합하고 있습니다.

핵심 역량

DB 최적화·동시성 제어·분산 시스템

- 100+ 컬럼 마케팅 테이블 정규화와 인덱스 튜닝으로 조회 지연 18초 → 0.5초 개선
- Deadlock 근본 원인 분석·분산 락 정립으로 데이터 계층 병목 해결
- Redis Streams 기반 WAL을 4-shard Valkey 클러스터와 serverless fanout 계층으로 고도화해 실시간 채팅 메시지 순서 보장·수평 확장 (수용률 99.39 → 99.99%, ack p95 767ms → 6.6ms)

서버리스 아키텍처 & 비용 최적화

- 배치 → EDA 전환으로 처리량 270배 확장, Lambda 하이브리드 파이프라인 도입
- BigQuery 과금 모델 분석 후 읽기 방식 전환으로 82% 비용 절감

레거시 현대화 & 안정성

- Express 단일 파일 → NestJS 3단계 점진적 전환, TDD 도입으로 주요 릴리스 무장애 배포
- 공통 패키지 체계 정립으로 7개 프로젝트 코드 일원화 (배포 3시간 → 15분)

개발 생산성·품질 검증 체계

- AI 지원 개발 결과를 TDD·통합 테스트·리뷰 체크리스트·Git 혹은 검증하는 팀 품질 게이트 정립
- Git 기반 공통 컨텍스트와 로컬 RAG/MCP 검색으로 코드 리뷰 전 도메인 지식·규칙 확인 자동화
- Games on AWS 2025 고객 세션에서 Amazon Q Developer 기반 데이터 파이프라인 확장 사례 발표

경력

게임듀오 — DEV팀 Platform Part Leader

2025.01 - 현재

모바일 게임 개발 및 퍼블리싱 회사. 6개 라이브 게임 환경(총 12 브랜치 운영) 공통 백엔드 플랫폼 설계, 데이터 파이프라인 구축, 공통 패키지 체계 운영

DEV팀 Platform Part Leader (2026.04 - 현재)

- Platform Part 신설 직후 공통 백엔드 과제의 우선순위와 완료 기준이 과제마다 달라 리뷰·배포 판단이 지연. 작업 유형별 완료 기준, 리뷰 체크리스트, 검증 템플릿을 정리해 플랫폼 과제를 백로그 → 검증 → 완료 흐름으로 표준화. 공통 인프라·패키지·채팅 서버 과제를 동일 기준으로 추적하고 리뷰 전 누락 항목을 줄이는 운영 체계 정착.
- AI 지원 개발 도입 후 코드 리뷰·테스트 기준이 개인별 사용 방식에 의존. AI 산출물도 일반 코드와 동일하게 TDD, 통합 테스트, 리뷰 체크리스트, Git 혹은 통과하도록 검증 기준을 정립. 도구 사용 여부보다 테스트 가능한 변경 단위와 리뷰 근거를 기준으로 산출물 품질을 판단하는 팀 흐름 마련.
- 팀 공통 코딩 표준·도메인 지식·검증 규칙이 개인 환경에 흩어져 신규/반복 작업의 편차 발생. 공통 규칙·템플릿·검증 혹은 Git repo 기반 설정으로 관리하고 로컬 검색/RAG로 참조 가능하게 구성. 반복되는 설정·검증 누락을 줄이고 코드 리뷰 전 확인해야 할 기준을 자동화된 체크리스트로 전환.

DEV팀 Server Developer (2025.01 - 2026.03)

- 마케팅 데이터 조회 비용 급증(전체 스캔 과금 구조). 파티셔닝 튜닝 분석 후 Write IOPS 증가 문제로 기각, gRPC 기반 분산 읽기(Storage Read API) 전환. BigQuery 조회 단가 기준 TB당 \$6.25 → \$1.1로 82% 절감.
- 배치 처리의 수집 기간 한계(60일). 배치 일괄 처리 대비 EDA 기반 온디맨드 처리로 전환 — 피크 부하 분산 및 처리량 확장. 수집 범위 6배 확장 (60일 → 360일), 처리 시간 2시간 → 5분 단축.
- 7개 프로젝트 간 코드 중복과 컨벤션 불일치. 수동 코드 동기화 대신 공통 NestJS 패키지 체계 도입 (유틸 10모듈 + 게임서버 Kit 자동 배포). 업그레이드 자동화로 배포 시간 3시간 → 15분 단축, 7개 프로젝트의 공통 코드 수정 지점을 단일 패키지로 일원화.
- 3개 광고 플랫폼 데이터 분산에 따른 마케팅 조회 지연(18초). 100개 컬럼 전체 스캔 회피를 위해 읽기 경로 정규화 + 인덱스 튜닝. 조회 성능 97% 개선(18초 → 0.5초).
- 게임 확률 공시 의무화 규제 리스크. DynamicModule 기반 확률 패키지를 다중 게임에 통합 + CDK 기반 감사 로그 파이프라인. 통합 테스트 커버리지 94%+ 확보(12 suites, 115 tests).

프로젝트: 실시간 게임 채팅 서버 (멀티테넌트), 마케팅 통합 플랫폼, 사내 공통 라이브러리 체계

제이앤피메디 — 개발2팀 Docs Squad Backend Engineer

2023.06 - 2024.08

누적 160억 투자 유치 임상시험 데이터 관리 솔루션 스타트업 (MSA 기반)

- 월 5건+ 발생하던 전자서명 처리 실패와 임상 데이터 정합성 리스크. FK 공유 락 교착 패턴을 재현·규명하고 트랜잭션 처리 경계를 조정해 동일 락 교착 재발을 방지. 전자서명 처리 실패 월 5건+ 해소.
- VDR 프로젝트 일정 지연 및 MVP 출시 압박. BE 2명 + FE 1명 팀으로 3주 긴급 투입, Event-Driven PDF 변환과 권한 기반 워크플로우 구현. MVP 출시 일정 준수.
- 이메일 발송 실패 사후 인지(수시간 지연). SES+SNS 이벤트 파이프라인 구축. 반송·바운스 감지 수시간 → 수초 단축, Slack 즉시 알림 체계 수립 (기술 블로그 게재).

프로젝트: Maven Docs 시스템 안정성 개선, Maven VDR MVP 개발

메드고 — 개발팀 Backend Engineer

2022.04 - 2023.06

누적 30만 다운로드 비대면 진료 및 약배달 플랫폼 스타트업

- Express.js 단일 파일(app.js) 중심 구조로 인증·예약·알림 기능 변경 시 회귀 위험 증가. 3단계 점진적 전환(단일 파일 → 레이어드 → NestJS) + JS → TS 마이그레이션. TDD 80% 달성, 주요 릴리스 구간 장애 0건 유지.
- 수동 처방전 입력 병목(약사 건당 3~5분). Naver Clova OCR + 식약처 API 연동으로 처방전 → 복약지도 자동화. 약사 처리 시간 90% 단축(3~5 분 → 30초).
- 진료·조제·배달 상태 반영 지연(수십 초). Socket.IO 실시간 동기화 + Web Push 알림 도입. 상태 반영 1초 미만으로 단축, 환자 대기 불만 해소.

프로젝트: 의사/약사 웹 서비스 고도화, 백엔드 아키텍처 전환

심플한 — 개발팀 Backend Engineer

2021.07 - 2022.03

고객 맞춤형 소프트웨어 솔루션 기업

- 학습 관리(출석, 수료, 진도율) 수작업 의존. LMS 폴스택 개발(QR 출석, PDF 유료증, Excel 일괄 등록). 학기당 300명+ 수강생 관리 전 과정 디지털화.
- 레거시 PHP 시스템에 신규 매칭·상담 기능을 추가해야 하나 기존 코드 변경 리스크가 큼. 독립 Spring Boot 앱을 제한된 iframe/postMessage 경계로 통합하고 가중 평균 매칭 알고리즘 구현. 레거시 영향 범위를 분리한 상태로 JOB Agent 서비스 출시.

프로젝트: 학습 관리 시스템(LMS) 개발

개인 프로젝트 (사이드) — 플랫폼 · 툴링 엔지니어링

2026.04 - 현재

직무 외 자기주도 플랫폼/툴링 엔지니어링 — 개발 워크플로우 자동화 프레임워크 및 사이드 프로젝트

프로젝트: 공개 AI CLI 런타임 런처

기술 스택

Backend: Node.js, TypeScript, JavaScript, NestJS, Express.js, TypeORM, Java, Spring Boot, Jest, TDD, REST API, gRPC, Testcontainers, Integration Test, OAuth2 — 레거시 → NestJS 점진적 전환, 공통 패키지 체계 정립, TDD 기반 개발, REST API / gRPC 설계

Database: MySQL, BigQuery, ElastiCache (Redis), Redis, Redis Streams, Valkey, ClickHouse — 쿼리 최적화, Deadlock 근본 원인 분석·해결, 대규모 테이블 정규화, Redis 분산 락

Cloud & Infra: AWS (Lambda, ECS/Fargate, S3, SQS/SNS, CDK, EventBridge, Kinesis, Athena), GCP (BigQuery, Pub/Sub), Docker, GitHub Actions, CI/CD, Terraform, ARM64 (Graviton) — 배치 → 서버리스 전환, EventBridge·Kinesis 기반 이벤트 파이프라인, BigQuery 비용 최적화, CDK 기반 인프라 코드화, CI/CD 파이프라인 설계

Quality & AI Verification: TDD Governance, Integration Test, Code Review, Git Hooks, Evidence Search, AI Output Verification — AI 도구 활용 결과를 TDD·통합 테스트·리뷰·Git 훅·근거 검색으로 검증하는 개발 품질 체계

학력

인하공업전문대학

학사 (학사학위전공심화, 재직 병행), 컴퓨터정보공학과

2022.02 - 2023.02

인하공업전문대학

전문학사 (재직 병행), 컴퓨터정보과

2017.02 - 2022.02

자격증

SQLD (Structured Query Language Developer) — 한국데이터산업진흥원, 2025.12

정보처리기사 — 한국산업인력공단, 2023.06

기술 공유

Amazon Q Developer를 활용한 개발 생산성 향상

외부 발표 — 2025.10

Games on AWS 2025 고객 세션 발표

AWS SES 이벤트 로그를 통한 이메일 발송 모니터링

사내 세미나 → 기술 블로그 — 2024.03

사내 세미나 발표 후 기술 블로그 게재

교육 프로그램

향해 플러스 백엔드 6기

2024.09 ~ 2024.11

향해99

TDD 및 클린 아키텍처, 동시성 제어, Redis 활용

1주차 명예의 전당 선정, 블랙 배지 수료

오픈소스

oh-my-openagent: Anthropic 모델 컨텍스트 제한 캐시 수정

2026.03

opencode: macOS sidecar 프로세스 hang 버그 수정

2026.02

CodexBar: kiro-cli 1.24+ Q Developer 포맷 지원

2026.02

학습 활동

시스템 디자인 스터디

2025.08 ~ 2026.04

가상 면접 사례로 배우는 대규모 시스템 설계 기초 1·2권

Node.js 디자인 패턴 바이블 스터디

2025.10 ~ 2025.12

Node 코어 개념, 동시성, 이벤트 루프, 모듈 시스템