

오현택 Backend Engineer

Email oht1312@gmail.com

Mobile 010-7126-4179

Github <https://github.com/HyunTaek5>

Blog <https://eight20.tistory.com>

LinkedIn <https://www.linkedin.com/in/dylanoh>



소개

- 3년차 백엔드 엔지니어로 누적 회원수 300만명의 에듀테크 서비스 프로젝트를 리드하여 런칭한 경험이 있고, NestJS, Spring Boot를 활용한 API 서버를 개발하였습니다. Github Actions, Jenkins를 이용한 CI/CD, AWS 인프라 구축을 하면서 서비스를 런칭한 경험이 있습니다.
- 반복되는 프로세스를 코드를 통해 자동화하여 프로젝트 세팅 및 배포에 소요되는 시간을 개선한 경험이 있습니다. 수동으로 프로젝트 배포하던 프로세스의 자동화를 통해 수동 배포하던 시간을 단축시켰습니다.

경력

Entropy Paradox / Backend Engineer (2021.05 ~ 2024.01)

웹 MVP 프로젝트를 End-To-End로 개발 및 서비스를 운영하였습니다. 사내에서 사용하던 서버 로깅 시스템을 Winston 기반 로그 시스템의 신규 구축을 통해 대체하고, Docker 다중 컨테이너 기반 무중단 배포 자동화를 구축하였습니다. 또한 이메일 발송 관리 시스템을 개선한 경험이 있습니다. Grafana, Prometheus 기반 APM, 인프라 모니터링 시스템을 구축하여 서비스의 운영 단계에서 모니터링 관제를 추가하였습니다.

트트영어 LIVE : 1대1 화상영어 서비스

초등학생을 대상으로 초등영어 콘텐츠를 1대1 화상으로 제공하는 누적 회원수 300만명의 에듀테크 서비스입니다. 느린 응답 속도 개선을 목표로 서버 안정화 작업을 진행하였고, 부하테스트와 모니터링 시스템을 구축하여 안정성을 향상시켰습니다.

- API 서버 애플리케이션 및 게이트웨이 구축**
 - 유저 중복 접속을 방지하기 위해 Socket.IO 도입
 - 온라인 결제를 위해 PG사 연동 결제 시스템 구축
- API 성능 개선**
 - 프론트에서 전체 데이터를 조회하여 처리하던 로직을 백엔드에서 정렬, 필터하여 조회 후 반환하는 방식으로 전환하였고, 한번에 최대 30개의 데이터만 조회가능하도록 페이징 처리하여 대시보드 화면의 로딩 시간 약 2초 개선 및 쿼리 성능 향상 (2100ms → 800ms)
 - 랭킹 조회시 랭킹 산정하던 로직을 5분 단위 스케줄링 작업의 별도 로직으로 분리하였고, Redis 캐시의 TTL을 5분으로 설정하여 특정 API 엔드포인트 호출시 부하가 걸리던 현상 개선 (서버 CPU 로드율 약 15% 개선)
- 모니터링, 알림 시스템의 구축을 통해 서버 부하 및 장애 발생 여부 확인 자동화**
 - 부하 시나리오 테스트를 통해 API의 성능 및 신뢰성 검증
 - 장애 발생시 연결된 슬랙 채널을 통해 부하, 장애 알림 발송

기술 스택 - TypeScript, NestJs, TypeORM, MSSQL, Window Server 2012, Redis(Cache), Nginx, Prometheus, Grafana, K6, Github Action, PageCall, Socket.IO

슈퍼스쿨 : 학교 생활 종합 플랫폼

학교 생활과 관련된 출결, 교과활동, 행정관리 작업을 지원하고, NICE와 연계된 대시보드를 제공하는 누적 학생 수 3만명의 에듀 테크 서비스입니다. 기존 레거시 코드의 느린 응답 속도 개선을 목표로 서버 안정화 작업을 진행하였고, 부하테스트와 추가적인 리소스 모니터링 시스템을 구축하여 가용성과 안정성을 향상시켰습니다.

- **API 서버 애플리케이션 구축 및 레거시 리팩토링**
 - 기존 레거시 코드 중 잦은 장애가 발생하던 부분을 REST API로 리팩토링
 - 불필요한 쿼리의 반복 실행을 방지하기 위해 Redis Cache 도입 (1000ms → 600ms)
- **학생 출결 데이터 시각화 대시보드 REST API 구현 및 레거시 성능 개선**
 - GraphQL에서 1:N 관계에서 발생하는 N+1 슬로우 쿼리에 Data Loader를 적용하여 1번의 조회 쿼리로 개선 (1700ms → 900ms)
 - 프론트에서 전체 데이터를 조회하여 처리하던 정렬/필터 로직을 백엔드에서 정렬, 필터하여 조회 후 반환하는 방식으로 전환하였고, 한번에 최대 30개의 데이터만을 조회가능하도록 페이징 처리하여 대시보드 화면의 로딩 시간 3초 개선 및 쿼리 성능 향상 (3700ms → 700ms)
- **모니터링 & 알림 시스템 구축을 통해 서버 부하 및 장애 확인의 자동화**
 - 부하 시나리오 테스트를 통해 API의 성능 및 신뢰성 검증

기술 스택 - TypeScript, NestJs, TypeORM, PostgreSQL, Redis(Cache), Nginx, Prometheus, Grafana, K6, Github Action

Banana Parts : 전기부품 포털 플랫폼

오프라인에서 발행되던 전기부품의 정보 및 뉴스, 부품 판매 및 견적을 제공하는 포털 서비스입니다. 판매자의 대량 부품 판매 등록 파이프라인의 목표 처리량 5000건을 안정적으로 처리하면서 서버의 고가용성을 유지하기 위해 부하테스트와 모니터링을 통해 안정성을 향상시켰습니다.

- **API 서버 애플리케이션 및 대량 부품 등록 파이프라인 구축**
 - 파이프라인 구조 : Server → MQ → Parsing → MQ → Validation → MQ → User Notice
 - Non-blocking, 순차 처리 신뢰성을 위해 Redis 기반 Bull Message Queue 도입
 - 알림 발송 서비스와 비즈니스 로직간의 결합성을 낮추기 위해 메시지 기반 유저 알림 비동기 발송 구현
- **실시간 인기 위키 랭킹 처리를 위한 Worker, REST API 구현 및 성능 개선**
 - 일별/시간대별 위키 조회수의 산정 범위를 30분 단위에서 10분 단위로 변경하여 랭킹 산정시 부하를 분산시키고, 인기순 위키 조회시 Redis 캐시의 TTL을 10분으로 설정하여 조회 성능 향상 (2500ms → 500ms)
 - 쿼리문에 Prepare Statement를 사용하여 쿼리 인젝션 공격 방어 및 옵티마이저 최적화
- **Jest, Mockito를 활용한 핵심 비즈니스 로직의 TDD 도입**
 - 유닛 테스트 및 E2E 테스트를 통해 핵심 비즈니스 로직의 신뢰성 검증
- **모니터링 및 장애 알림 시스템 구축을 통한 API 서버 부하 및 장애 슬랙 알림 자동화**

- 부하 시나리오 테스트를 통해 파이프라인과 API의 성능 및 신뢰성 검증

기술 스택 - TypeScript, NestJs, TypeORM, MySQL, Jest, Mockito, Redis(BullMQ, Cache), Nginx, Prometheus, Grafana, K6, Github Action, NHN Cloud(SMS, Email, 알림톡)

기술

- Language : TypeScript, Java, Kotlin, Python	- Server : NestJs, ExpressJs, Spring Boot, Django, Flask
- Database : MySQL, PostgreSQL, MSSQL	- ORM : TypeORM, Prisma, Spring Data JPA
- Cache : Redis	- Monitoring : Prometheus, Grafana
- MQ : BullMQ, RabbitMQ	- Logging : PromTail, Loki
- Client : NextJs, React, Tailwind CSS	- Infra : AWS (EC2, S3, RDS, SQS, ECR, ELB, Cloudfront, Amplify)
	- ETC : Nginx, PM2, K6, Github Actions

프로젝트

[그님레](#) WoT 전적/레이팅 시각화 서비스/ Full Stack (2022.10 ~ 서비스 운영 중)

월드 오브 탱크(World of Tanks) 유저의 전적 데이터, 레이팅 관련 수치를 시각화하여 제공해주는 서비스입니다. 디스코드 봇을 통해 서비스를 제공 중이며, DAU 100명 정도의 트래픽이 발생하고 있습니다.

- **NestJs를 사용하여 API 서버, 배치 애플리케이션 및 레이팅 계산 파이프라인 구축**
 - 전적 데이터 기반 커뮤니티 비공식 레이팅(WN8) 계산 파이프라인 구축
 - 파이프라인 구조 : Server → MQ → Wargaming API → Parsing → MQ → Calculating → Client Webhook
 - 중복 유저 전적 조회시 발생하는 부하를 분산시키기 위해 Redis 캐시의 TTL 15분으로 설정하여 조회 성능 개선(4500ms → 1000ms)
- **Discord.js 기반 디스코드 봇 클라이언트 구현**
 - API 서버. 배치에 장애가 발생하여 전적 데이터 계산에 시간이 오래 소모되는 경우에도 TimeOut 응답이 아닌 서버에서 발송한 웹훅을 수신받아 유저에게 제공할 수 있도록 웹훅 수신 구현
- **도커 다중 컨테이너 기반 온프레미스 환경 배포**
 - 개발 생산성의 향상을 위해 무중단 배포 자동화 구축, Rolling 배포방식의 CI/CD 도입
- **APM | 인프라 모니터링 및 장애 알림 시스템 구축을 통한 서버 부하 및 장애 알림 자동화**

기술 스택 - TypeScript, NestJs, TypeORM, PostgreSQL, Jest, Mockito, Redis(BullMQ, Cache), Nginx, Prometheus, Grafana, K6, Github Action, Docker, Docker-compose, Discord.js

Mask On 매장별 마스크 착용 현황 제공 서비스 / Backend (2021.01 ~ 2021.03)

코로나-19 사태로 인한 매장내 마스크 착용률 정보를 5분마다 측정하여 매장별로 정리하여 유저에게 제공하는 서비스입니다.

- Django를 사용하여 유저, 매장, 마스크 착용률 도메인 기능 개발
 - Django/ Docker/ Docker-compose를 활용하여 API 서버 개발
 - RaspberryPi에서 OpenCV와 YOLO v5를 통해 이미지에서 산출한 로그 및 센서 정보 서버 송수신 구현
 - Postman 기반 API 문서화
- Docker 컨테이너 기반 AWS EC2 배포
 - 수동 배포 시간의 절약을 위한 배포 자동화, Blue/Green 배포방식의 CI/CD 도입
 - Grafana, Prometheus 기반 도커 컨테이너 리소스 모니터링 도입

기술 스택 - Python, Django, Django Rest Framework, MySQL, Nginx, Gunicorn, AWS, Grafana, Prometheus, Github Actions

활동

Techer [테커] 2021.03 ~

- 실리콘밸리 개발자의 SW 개발자 커리어 그룹
- 실리콘밸리 한달 살기, 기술 컨퍼런스, 해커톤, 네트워킹 행사 등 다수 경험

멘토

- [테커 파트너스] Backend 파트 멘토 2022.02 ~ 2022.08
- [한국공학대학교 x 테커] 2023 실리콘밸리 개발자 아이디어 해커톤 2023.05.26 ~ 2023.05.27
- [D.camp x 티타임즈 x 테커] 2023 테커 하계 부트캠프 쿼리 튜닝 세션 2023.07.28

스터디 팀장

- 자바 ORM 표준 JPA 2023.04 ~ 2023.07
- 어쩌다 Nest 2023.11 ~ 진행중

부트캠프 참가

- AI 실리콘밸리 헤드스타트 인턴십 부트캠프 1기 수료 2021.01 ~ 2021.02

교육

단국대학교 / 모바일시스템공학과 2020.03 ~ 현재