

김윤영

“다차원적으로 성장하는 한계없는 개발자”

✉ yunryy329@gmail.com

💻 <https://github.com/yunrry>

🌐 <https://yunrry.github.io/>

Key Point

Spring 기반 서버 개발 및 AI·DevOps 경험을 바탕으로,
실제 서비스 구축과 운영에 강점을 가진 백엔드 개발자입니다.
AI 모델과 데이터 처리 기술을 서비스에 통합하여
지능적이고 효율적인 시스템을 구현하는 데 집중하고 있습니다.

자동화·운영 효율·협업을 통해 서비스 품질과 팀 생산성을 함께 성장시키는
개발자가 되겠습니다.

Check Point

- ✓ 쿠버네티스 환경에서 ArgoCD, Jenkins를 통한 빌드·배포 경험 보유
- ✓ 데이터 벡터화 & 추천 알고리즘 구현 경험
- ✓ LLM API를 활용한 프롬프트 설계 및 요청 최적화
- ✓ App 서버, Batch 서버, DB, 로드밸런서, Redis, Kafka를 포함한 서버 아키텍처 설계
- ✓ React + TypeScript 프레임워크로 프론트엔드 개발 가능
- ✓ 애자일 기반 협업과 적극적인 소통 중시

Skills

Spring Boot AWS Docker Kubernetes MySQL Redis Git React

Resume

Profile



김윤영

Etc.

2025.02 제주대학교 컴퓨터공학 졸업

제 14회 구름톤 in Jeju 최우수상

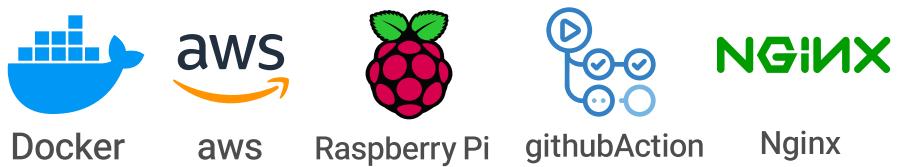
카카오(kakao) 현장실습 인턴 1개월

소프트웨어학부 학생회 학술차장, 전시회 기획

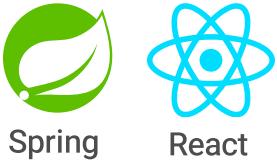
북경이공대(BIT) 국제 여름 공학 프로그램 참여

Skills

DevOps/Infra



FrameWork



Data



Test/API docs



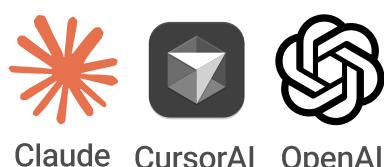
Monitoring/Logging



Management



AI tools



Language



경험 1회



10 Projects

- ▶ FLIK - RAG 기반 AI 여행 코스 추천 서비스 개발
- ▶ 도르멍드르멍 - 제주 오디오 스토리텔링 관광 서비스 개발
- ▶ SafeWalk - 보행자 사고 안전 관리 플랫폼 개발
- ▶ 지역 소상공인 AI 홍보글 생성 서비스 개발
- ▶ 인스타그램 클론 프로젝트
- ▶ sns 대용량 트래픽 처리 시뮬레이션 프로젝트
- ▶ 웹소켓 택시 배차 구현 프로젝트
- ▶ 온라인 쇼핑몰 서버 성능 개선 & 모니터링 시스템 구축 프로젝트
- ▶ AI 동화책 생성 서비스 개발
- ▶ 식물 센서 데이터 수집 & 생장 예측 프로젝트

RAG 기반 맞춤 여행 코스 추천 서비스 • 플릭 FLIK

1인으로 관광데이터 공모전을 신청하고 엄두가 안나서 취소하려던 찰나,
좋은 기획자 팀원을 만나서 본 프로젝트를 도전하게 되었습니다.
제출 기한까지 1개월 남짓 남은 시간에 몰입하여 배포를 완료하였고
서비스 요구사항의 AtoZ를 혼자 구현해보는 역량을 쌓을 수 있었습니다.

핵심 기능

- ▶ RAG 기반 AI 코스 생성
- ▶ 플랜 저장 및 공유
- ▶ 여행기 작성
- ▶ 동선 지도 시각화

Skills

- Spring Boot
- AWS
- Docker
- RaspberryPi
- MySQL
- Redis
- Git
- React

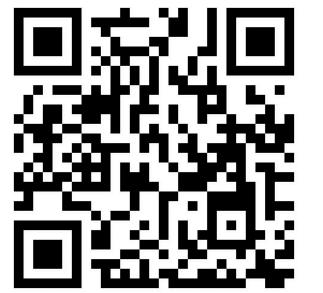
Team 2인구성 (기획/디자인 1, 풀스택 1)

Role 풀스택

Work 2025.08 - 2025.10(2개월)



<https://flik-tau.vercel.app/>



설계 및 구현

① App/Batch 서버 분리 구조

App Server: 실시간 API 요청 처리 (스와이프, 검색, 코스 조회)

Batch Server: 스케줄 기반 백그라운드 작업 (벡터 재계산, 통계 집계, 데이터 수집 및 동기화)

목적: 리소스 격리를 통한 API 응답 안정성 확보 및 배치 작업으로 인한 성능 저하 방지

성과: App 서버 CPU 사용률 30% 감소, API 응답 시간 안정화 (p95 200ms 이하 유지)

② 이중 DB 구조 (MySQL + PostgreSQL)

MySQL: 트랜잭션 중심 비즈니스 로직 (사용자, 장소, 코스, 게시글 CRUD)

PostgreSQL: 벡터 연산 전용 (pgvector 확장, 코사인 유사도 계산)

목적: 트랜잭션 처리와 대용량 벡터 연산 워크로드 분리로 각 DB 최적

성과: 벡터 검색 속도 40% 향상, MySQL 락 경합 70% 감소

③ Raspberry Pi 기반 Docker 환경

구성: App, Batch, MySQL, PostgreSQL, Redis, Nginx (총 6개 컨테이너)

리소스 최적화: JVM 힙 메모리 512MB 제한, Connection Pool 튜닝

가동률 99.8% 유지 (30일 기준 모니터링)

④ Blue/Green 무중단 배포

GitHub Actions: 코드 푸시 → 테스트 → 도커 이미지 빌드 → Green 컨테이너 배포

Nginx 스위칭: Health Check 성공 후 트래픽 전환 (다운타임 0초)

롤백 전략: 배포 실패 시 Blue 컨테이너로 자동 복귀

Discord Webhook: 배포 상태 실시간 알림

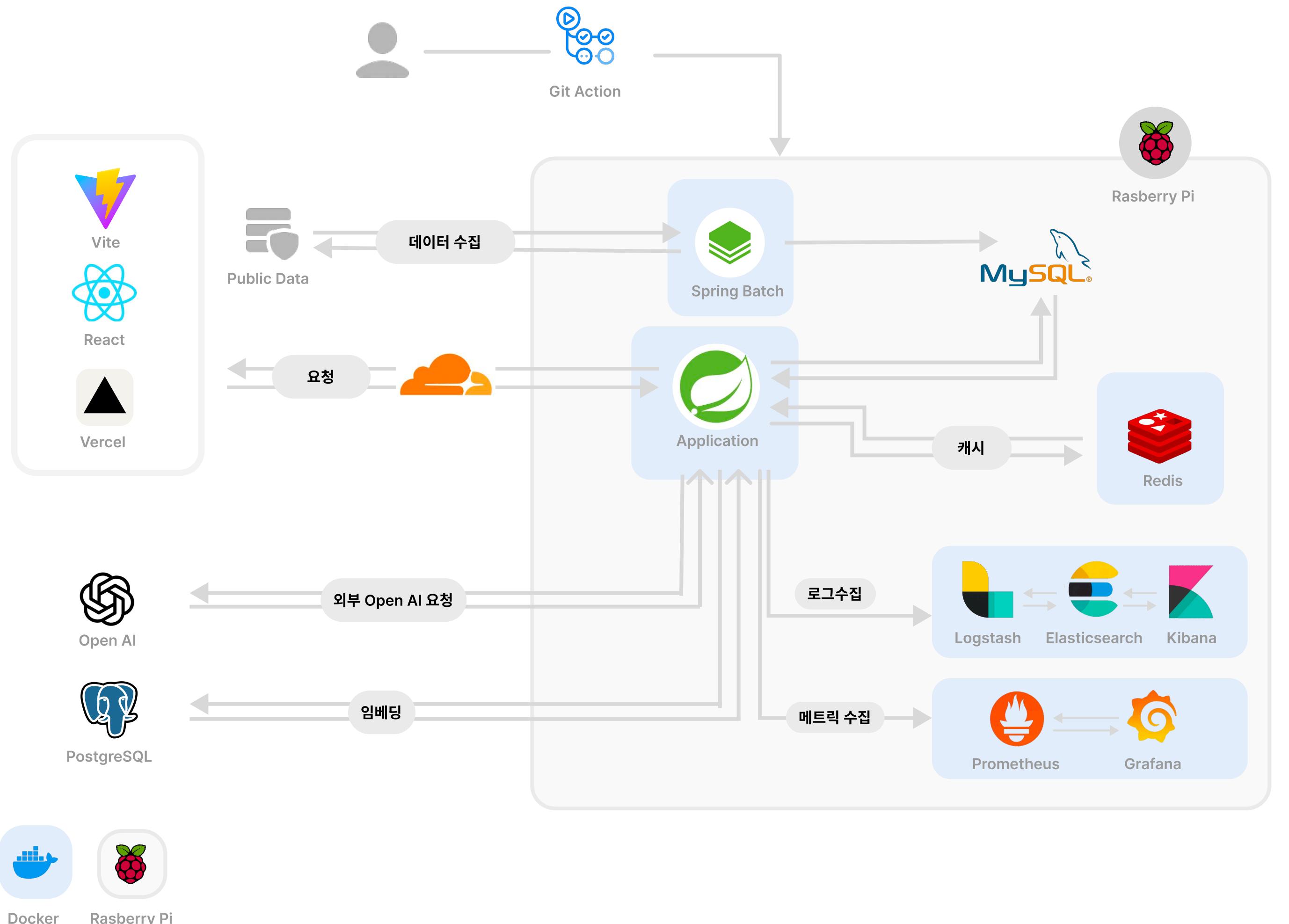
⑤ 네트워크 및 보안

Cloudflare: SSL/TLS 인증, DDoS 방어, CDN 캐싱

Nginx: 리버스 프록시, 로드밸런싱, Upstream 헬스체크

CORS 정책: 도메인별 허용 메서드 및 헤더 관리

성과: SSL 인증서 자동 갱신, 글로벌 엣지 네트워크 활용으로 응답 속도 개선



설계 및 구현

⑥ RAG 기반 추천 시스템

OpenAI Embedding API 활용

구현: 장소 텍스트 데이터(이름, 설명, 리뷰)를 1536차원 벡터로 변환
 저장: PostgreSQL의 vector 타입으로 인덱싱 (HNSW 알고리즘)
 검색: 코사인 유사도 기반 유사 장소 추출

실시간 사용자 선호도 학습

이벤트 기반 갱신: 사용자 스와이프 → Spring Events로 비동기 처리 → 선호도 벡터 업데이트
 알고리즘: 저장한 장소들의 평균 벡터를 사용자 선호도 벡터로 계산
 카테고리별 분리: 음식점, 관광지, 숙소 등 카테고리별 선호도 벡터 독립 관리
 성과: 추천 정확도 18% 향상 (사용자 피드백 기반 측정)

여행 코스 자동 생성 알고리즘

1. 사용자 선호 벡터 기반 후보 장소 추출 (카테고리별 Top-K)
2. 지역 좌표 기반 반경 검색 (10km → 20km → 30km 확장)
3. Haversine 공식으로 장소 간 거리 계산 및 최적 동선 구성
4. 필터 적용: 계절성, 영업시간, 접근성, 평점
5. 슬롯별 장소 배치 (조식/중식/석식/관광지 구분)

⑦ 이벤트 기반 아키텍처

Spring Events 활용

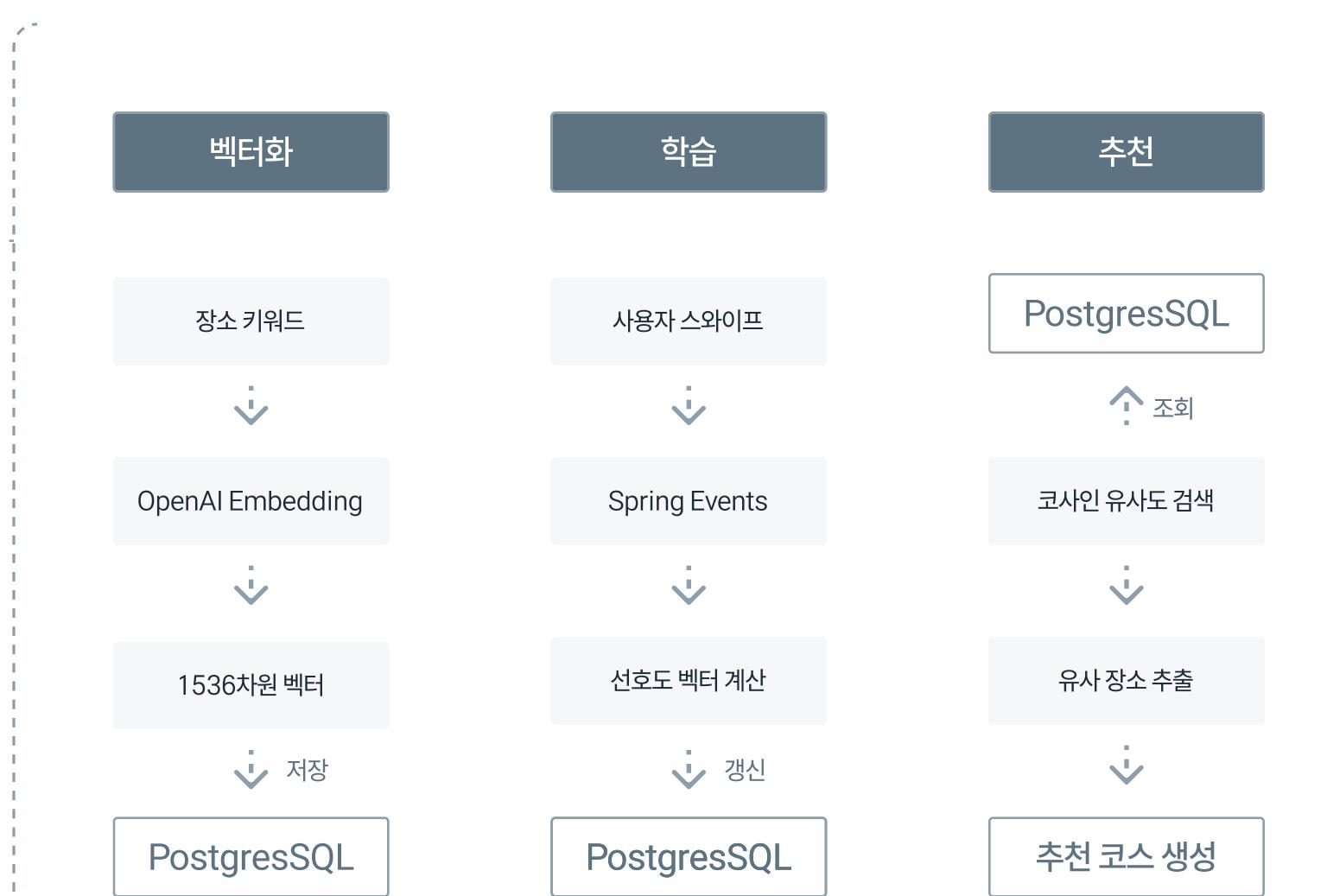
구현: 스와이프 저장 시 @EventListener로 4개 독립 작업 비동기 처리
 성과: API 응답 속도 50% 개선 (200ms → 100ms)

Redis Streams 도입

목적: 이벤트 영속성 보장 및 Kafka 마이그레이션 대비 인터페이스 설계
 구현: Consumer Group 패턴, Dead Letter Queue, 재처리 메커니즘
 성과: 서버 재시작 시에도 이벤트 유실 방지, 초당 1000+ 이벤트 처리 가능

⑧ 클라이언트 구현

구현: React + TypeScript SPA로 클라이언트 전체 기능 구현
 내용: 상태관리, 라우팅, Type안정성, 반응형UI, 스와이프 액션, API 호출 등



확장 가능한 구조 설계



핵심 성과/역량 요약하기

- ✓ 대용량 데이터 처리 실무 경험: 하루 5,000+ 데이터 수집, 30,000+ 벡터 검색
- ✓ 분산 시스템 설계 능력: 이중 DB 구조, 서버 분리, 이벤트 기반 아키텍처
- ✓ 성능 최적화 경험: API 응답 50% 개선, 벡터 검색 40% 향상, 락 경합 70% 감소
- ✓ Kafka 시스템 확장 가능성: Redis Streams → Kafka 전환 설계 완료
- ✓ 추천/광고 도메인 이해: RAG 기반 개인화 추천, 피쳐 집계 경험
- ✓ 프론트엔드 업무 이해도: 클라이언트 구현 경험으로 협업시 원활한 소통 가능

오디오 스토리텔링 기반 제주 여행 서비스

도르멍 드르멍

해커톤 주최측으로부터 제공된 쿠버네티스 환경에서 팀 전용 네임스페이스를 활용하여 내부 ClusterIP Service를 통해 Application과 DB 간 통신 구조를 구현했습니다. 이로써 해커톤 현장의 제한된 리소스로 외부 인프라 의존 없이 완전한 백엔드 서비스를 빠르게 구현/배포하는 경험을 가질 수 있었습니다.

핵심 기능

- ▶ 관광지 안내판에 부착된 QR 코드 스캔
- ▶ AI TTS(제주 방언) 스토리 생성
- ▶ 이야기 조각 수집

Skills

Spring Boot AWS RDS Docker Kubernetes MySQL LLM Git TTS

Team

5인 구성 (기획1, 디자인1, 프론트엔드2, 백엔드1)

Role

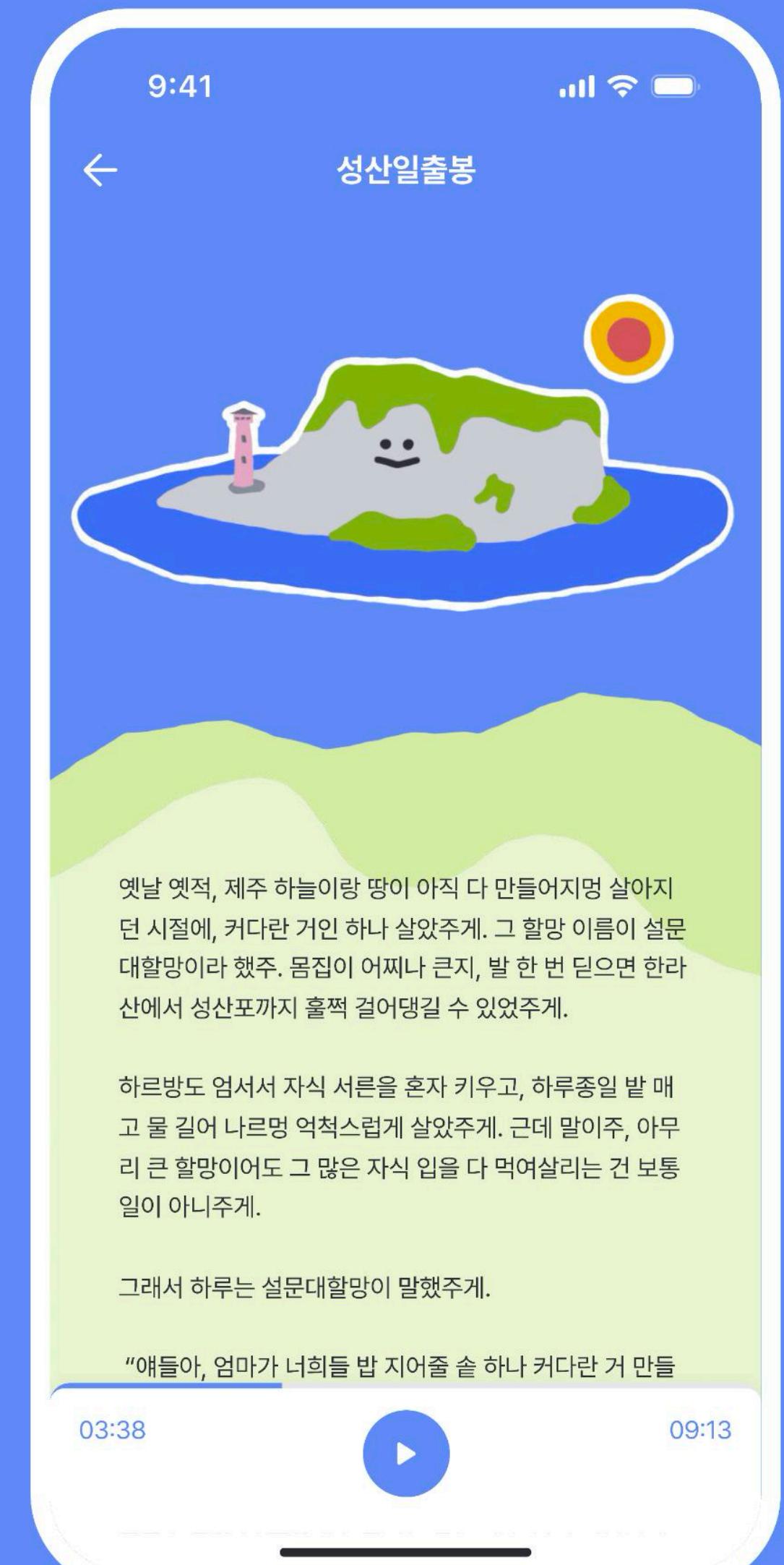
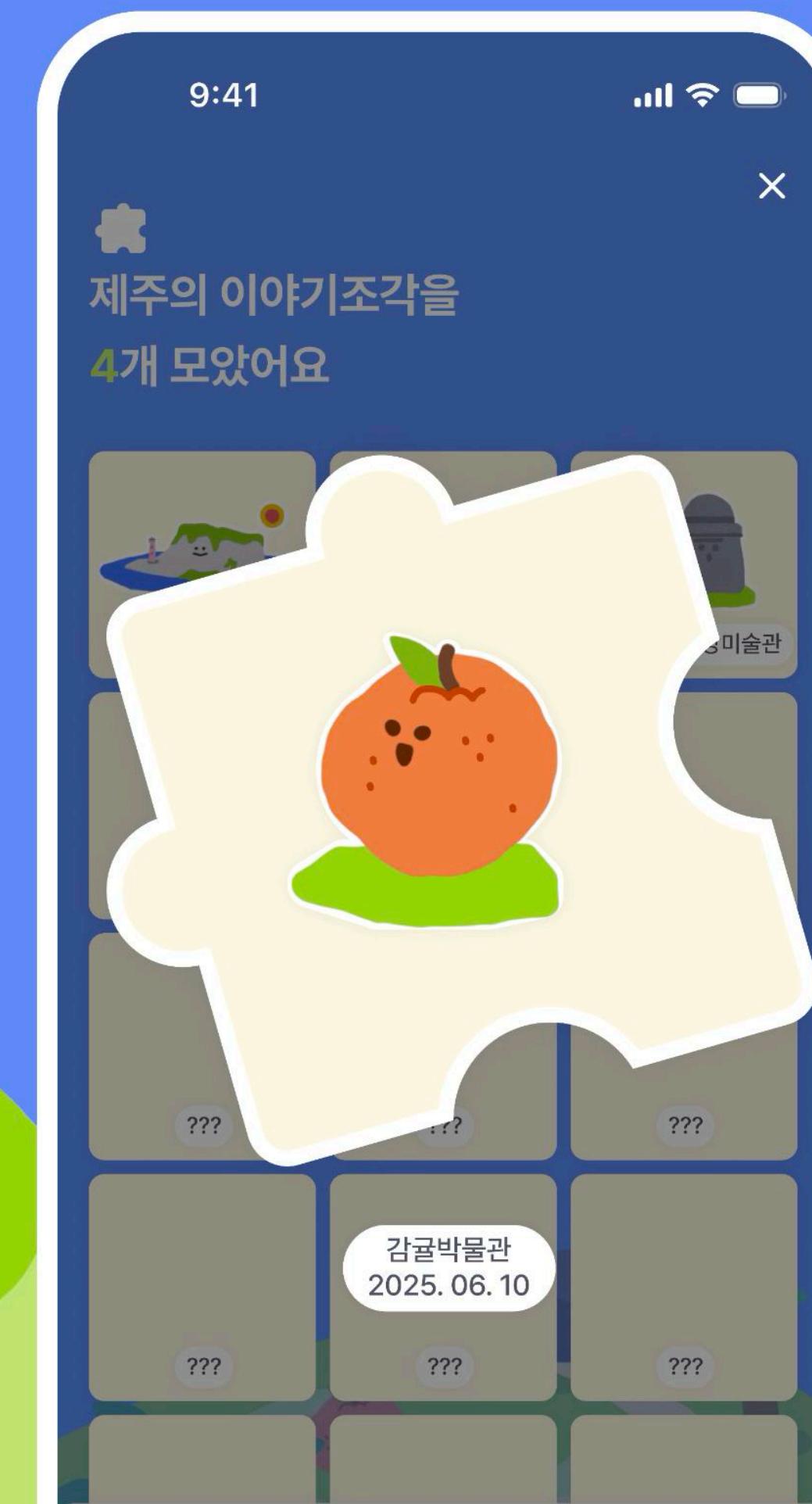
백엔드

Work

2025.07 (2박 3일 해커톤)



<https://dormung.netlify.app>



설계 및 구현

① Kubernetes + ArgoCD + Jenkins

CI/CD 파이프라인 구축 : Jenkins 빌드 → Docker 이미지 푸시 → ArgoCD 배포 자동화

제한된 권한 환경에서 Manifest 기반으로 배포 설계

환경: 대회 측에서 제공한 제한된 Kubernetes 클러스터

리소스 제한:

- ConfigMap, Secret, Ingress 접근 불가
- Pod, Deployment 삭제·조회만 가능

② Pod 분리 구조

Application Pod ↔ DB Pod (MySQL) 분리 운영

Namespace 내에서 Service로 내부 연결하여 통신

목적

- 외부 인프라 의존 없이 제공된 인프라만 사용
- 자원 격리 및 장애 대응 유연성 확보

전략: Persistent Volume으로 DB 데이터 영속화

③ 트러블 슈팅

문제: 새로 빌드된 이미지가 Pod에 반영되지 않음

시도

1. Pod 삭제
2. Deployment 재적용
3. Force Sync (ArgoCD)

원인: ConfigMap/Secret 접근 불가로 인해 Rollout Trigger 미작동

해결

- 접근 가능한 범위 내에서 Application Pod 완전 재생성 방식으로 문제 해결
- 문제를 구조적으로 분석 후 재현 및 문서화

④ 서비스 지속화 및 인프라 마이그레이션

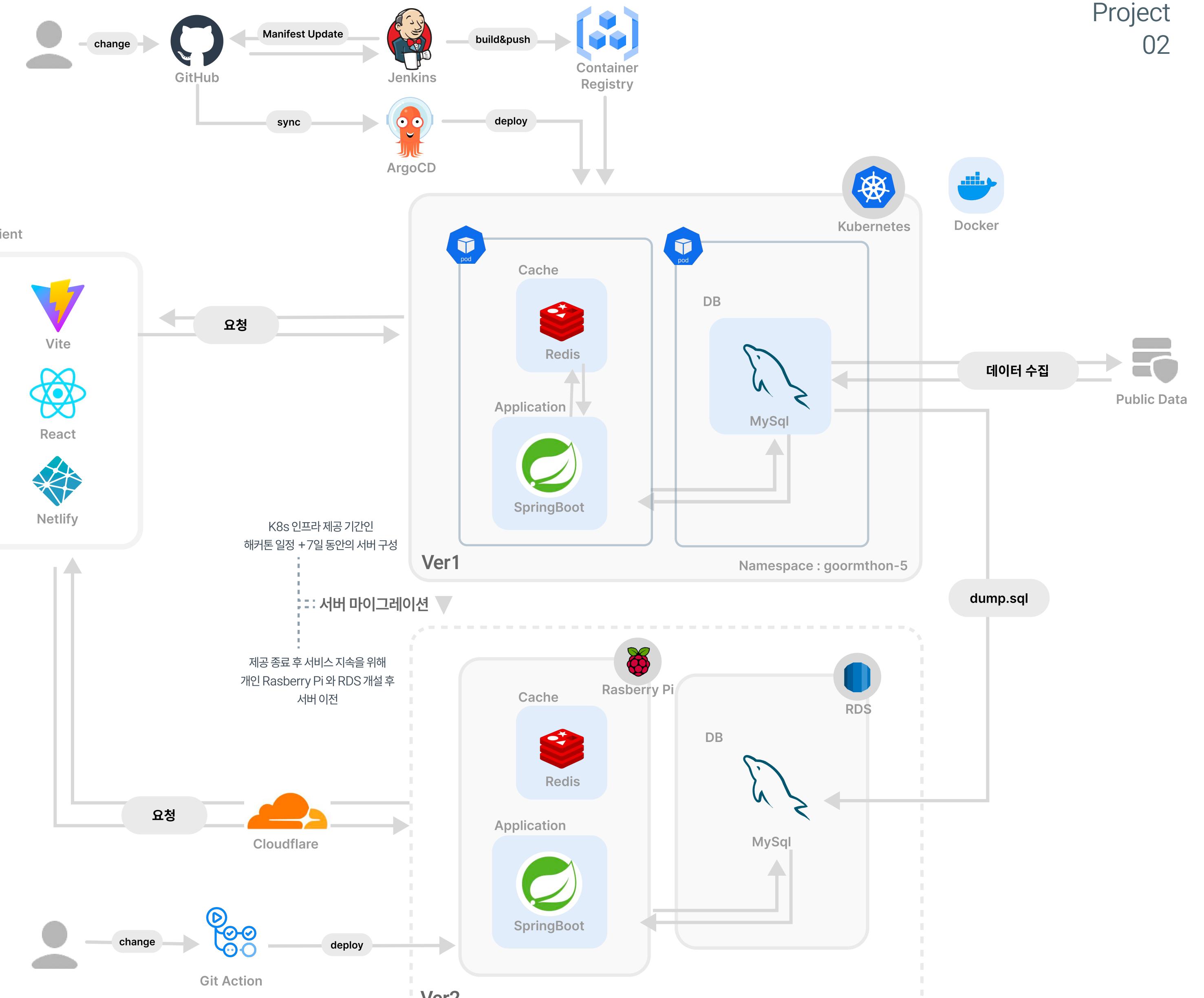
인프라 제공 종료 후 RDS + Raspberry Pi 환경으로 이전

DB: MariaDB → AWS RDS (MySQL) 마이그레이션

App: Raspberry Pi에서 Spring Boot 앱 구동 및 Nginx 리버스 프록시 구성

Network: Docker 환경 재구성, 포트 포워딩 및 SSL 리다이렉션으로 보안성 확보

CI/CD: Github Action으로 간소화된 자동 배포 유지



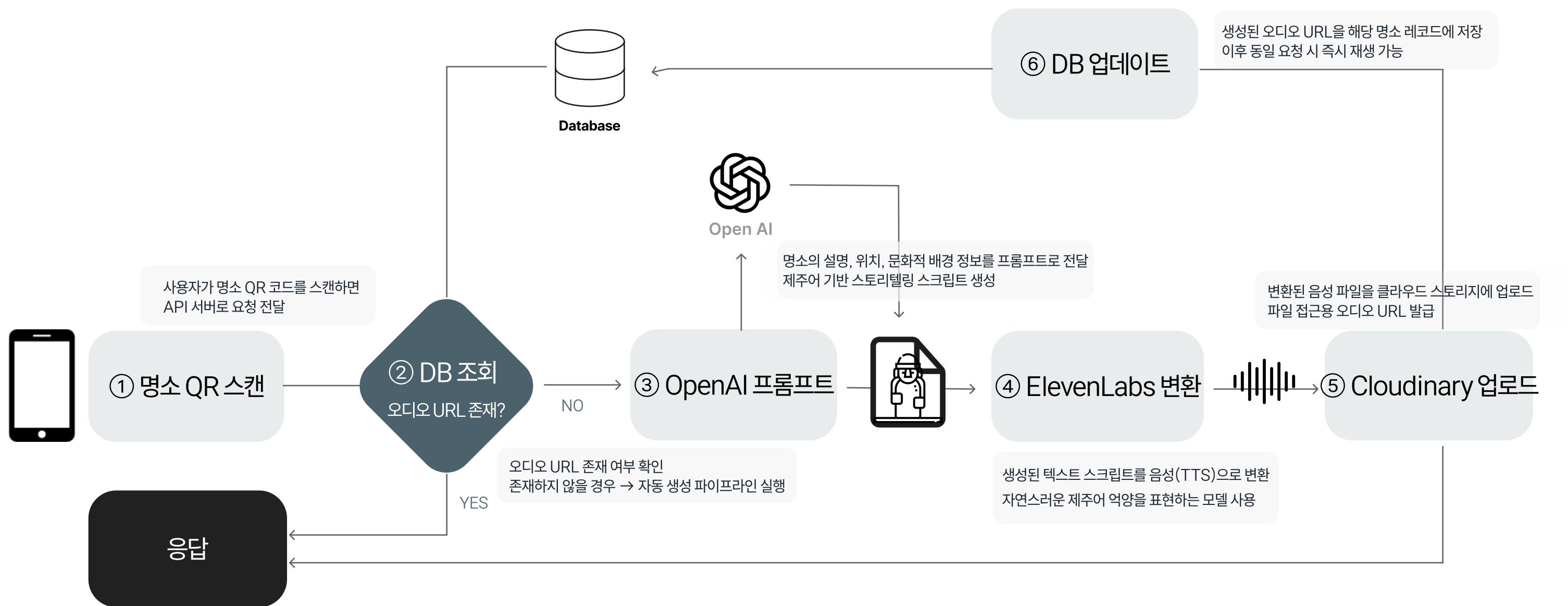
설계 및 구현

⑤ AI 스토리텔링 오디오 자동 생성 파이프라인 구현

목표: 명소 QR 스캔 시, 해당 명소의 오디오 콘텐츠가 존재하지 않는 경우
 → 자동으로 스토리텔링 스크립트를 생성하고 → 제주어 음성으로 변환하여
 → 클라우드에 저장하는 AI 기반 자동 생성 파이프라인을 구축.

⑥ 오디오 자동 생성 파이프라인 흐름

1. 명소 QR 스캔 요청 발생
2. DB 조회 및 조건 분기
3. OpenAI API 연동
4. ElevenLabs API 연동
5. Cloudinary 업로드
6. DB 업데이트



⑦ 성과 및 특징

- 오디오 파일 생성 및 관리의 완전 자동화 달성
- 스토리텔링 콘텐츠 생성 시간 수작업 대비 90% 이상 단축
- 제주어 기반 음성 생성으로 지역성 있는 사용자 경험 강화
- DB 연동 및 캐싱 처리로 중복 요청 최소화 및 서버 효율성 개선



⑧ 협업

- 5명의 팀원들과 현장에서 처음 만나 2박3일만에 MVP 완성
- 효과적인 의사소통과 배려로 빠른 협업 능력 체득
- 작업 현황 공유
 - 방향성 설계, planB 까지 대책 설정
 - 자신의 우선순위 뿐만 아니라 상대의 우선순위 파악
 - 요구사항 파악

- ✓ 운영환경 제작 속에서 문제 해결 프로세스를 체득
- ✓ 실무형 DevOps 역량 강화
 - Kubernetes, ArgoCD, Jenkins 연동 원리 이해
 - 클라우드(RDS) + 온프레미스(Raspberry) 혼합 아키텍처 직접 구현 경험
- ✓ 서비스 종료 후에도 독립적 운영 가능한 실무형 배포 구조 완성
- ✓ AI를 활용한 TTS(Text to Speech) 오디오 파일 생성 파이프라인 설계

Thank you

다차원적으로 성장하는 한계없는 개발자, 김윤영

지난 3년 동안 다양한 프로젝트 과정을 즐기고 배우면서 백엔드 개발자 역량을 쌓았습니다.

인간이 창의성과 본질에 집중할 수 있게 하는 시스템을 만드는 개발자가 되고싶습니다.

yunryy329@gmail.com

010 • 3494 • 1463