

9개 검색 소스를 격리하고, Kafka 9종의 복구를 좁히며, 오류 데이터 약 200만 건을 바로잡았습니다

Java·Kotlin·Spring으로 중단 뒤 다시 시작할 수 있고, 데이터가 맞는지 끝까지 검증할 수 있는 백엔드를 개발합니다. 채용 피드·이벤트 복구·포인트 원장·JPA 보정 사례를 문제-설계-검증 결과 순서로 정리했습니다.

서민재 · Rex Seo

Backend Engineer · Java / Kotlin / Spring

010-8719-3490

ckdekrn88@gmail.com

GitHub

Velog

LinkedIn

이력서

경력기술서

01

JOBKOREA·Albamon 개인화 피드의 9개 검색 소스를 독립 변경 단위로 분리하고, 캐시 비가용에도 후보 응답 경로를 유지했습니다

9개 검색 소스의 정책·실패 처리와 OpenSearch→Redis 경계를 분리했습니다. Hit·Empty·Unavailable 3개 Snapshot 상태에 따라 저장 결과 반환·새 Snapshot 저장·저장 생략 후 재계산 경로를 각각 구현했습니다.

01 · PROBLEM

검색 소스마다 정책과 실패 조건이 달랐습니다

02 · DECISION

후보 처리 단계를 명시적인 계약으로 나눴습니다

03 · BOUNDARY

3개 Snapshot 상태로 캐시 경계를 드러냈습니다

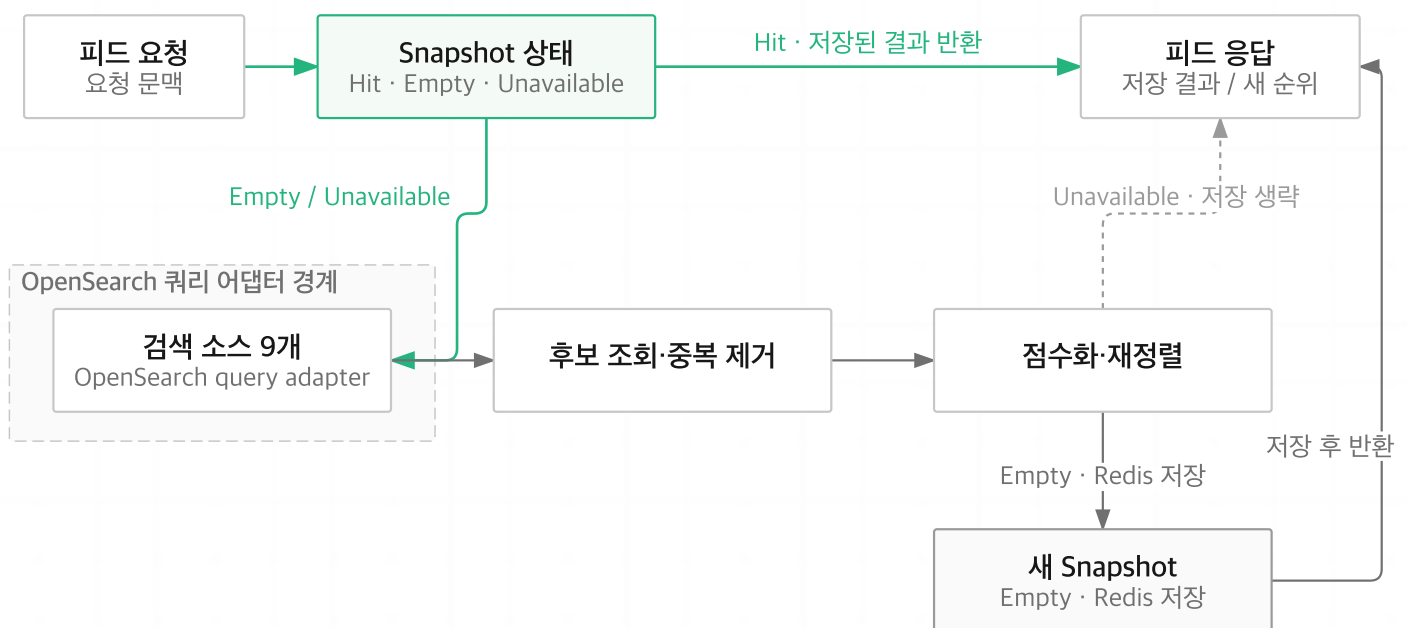
04 · VERIFICATION

검색 구현과 도메인 경계를 함께 지켰습니다

도식의 실선은 기본 조회·저장 경로, 점선은 캐시를 사용할 수 없을 때 저장만 건너뛰는 우회 경로를 뜻합니다.

9개 검색 소스에서 Snapshot 응답까지

Snapshot 상태가 달라도 후보 조회·점수화·재정렬의 경계는 유지됩니다.



Kafka 9종에서 미완료 업무만 선택 복구했습니다

담당	구현	저장	알림
Kafka 9종·상태 6개·복구 경로 8개·15분 정제 알림 설계	Kafka 발행·처리 실패를 6개 상태로 구분	PostgreSQL에 처리 상태 기록 · JSONB에 복구용 payload 저장	15분 정제된 IN_PROGRESS 이벤트를 유형별 Slack 알림
상황 같은 실패 상태로는 복구 지점을 정할 수 없었습니다 발행 전에 멈췄는지, 발행 후 업무 처리에서 멈췄는지에 따라 확인할 데이터와 재실행 범위가 달랐지만 하나의 실패 상태만 남았습니다.		바꾼 점 실패 단계와 업무를 분리해 복구 범위를 좁혔습니다 이벤트 9종을 6개 상태로 기록하고 payload를 보존했습니다. 8개 복구 경로에서는 차감·회수·만료 알림 중 미완료 업무만 재실행했습니다.	

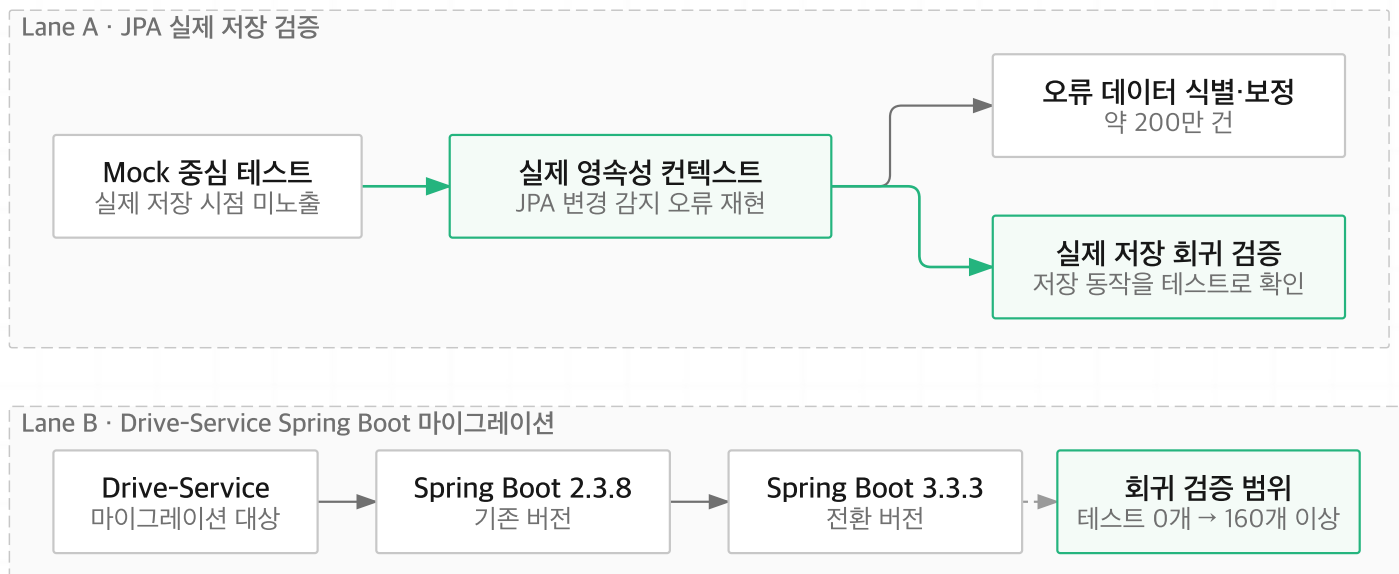
토스랩 JANDI에서 실제 영속성 컨텍스트로 JPA 저장 오류를 재현하고 약 200만 건을 보정했습니다

Mock 테스트의 사각지대와 실제 영속성 컨텍스트 재현을 구분했습니다. 데이터 보정과 실제 저장 회귀 검증도 각각 독립된 근거로 남겼습니다.

MOCK BLIND SPOT	PERSISTENCE REPRODUCTION	CORRECTION BOUNDARY	CORRECTED DATA	STORAGE REGRESSION
Mock 테스트는 실제 저장 경계를 지나지 않았습니다	실제 영속성 컨텍스트에서 저장 과정을 재현했습니다	영향 데이터 식별과 보정 실행을 분리했습니다	오류 데이터 약 200만 건을 바로잡았습니다	실제 저장 동작을 회귀 검증에 포함했습니다
INDEPENDENT WORK · MIGRATION Drive-Service의 버전 전환을 별도로 수행했습니다 Spring Boot 2.3.8 → 3.3.3으로 전환하며 회귀 검증을 위해 테스트를 0개 → 160개 이상으로 보강했습니다.				

Mock 사각지대 · 영속성 재현 · 데이터 보정 · 저장 회귀 검증

확인된 네 근거의 관계를 나타내며 정확한 수행 순서를 뜻하지 않습니다.



10개 테이블 배치를 재개하고, 9개 API·13개 시나리오로 포인트 정합성을 검증했습니다

10개 테이블 배치 재시작, 9개 API·13개 시나리오의 동시성 검증, Kafka 8개 선택 복구 경로를 서로 독립된 운영 단위로 나눴습니다.

LANE A · BATCH RESTART

배치별 진행 위치로 중단 지점부터 재개했습니다
포인트 소멸 대상을 페이지 단위로 처리하고, 만료 알림 대상은 별도로 수집했습니다.

LANE B · INVARIANT CHECK

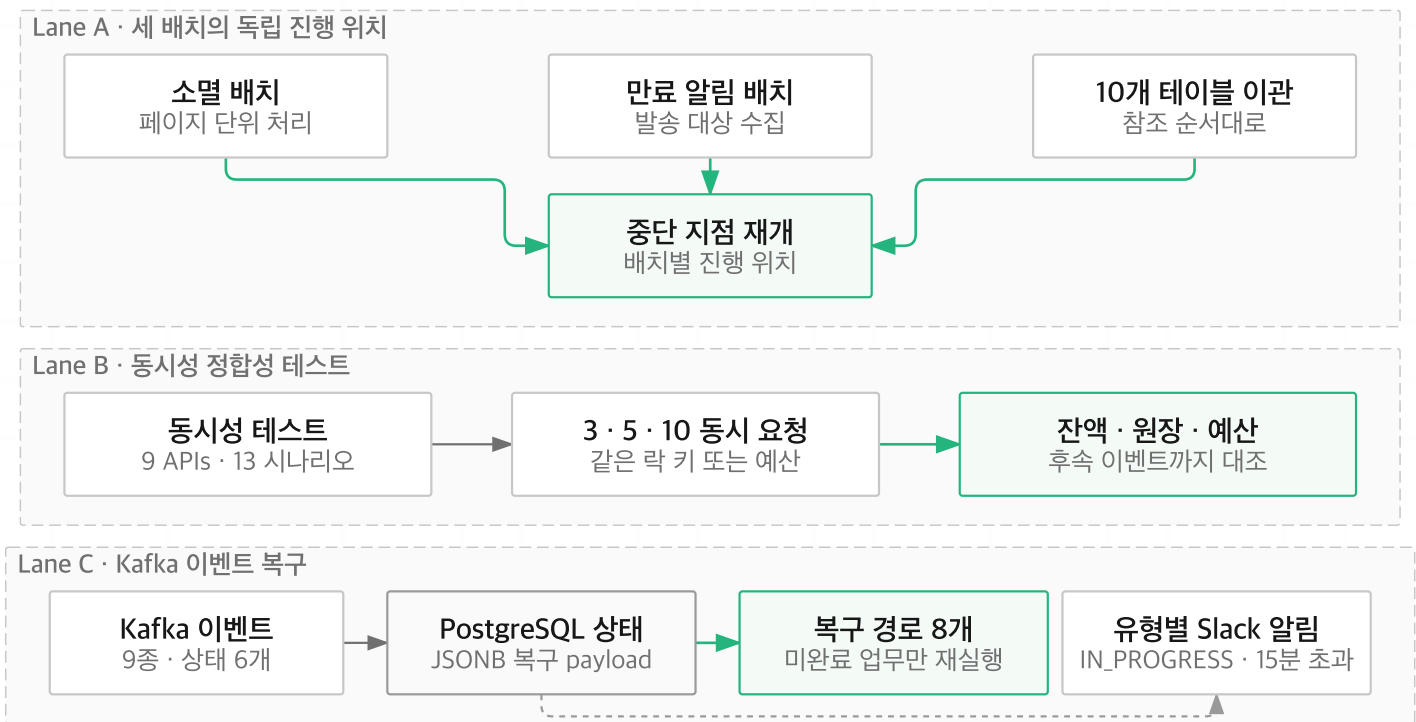
동시 요청의 결과 불변식을 확인했습니다
9개 API의 13개 동시성 시나리오에서 3·5·10개 요청을 같은 락 키 또는 같은 예산 조건으로 실행하고, 종료 뒤 잔액·원장·예산·후속 이벤트를 함께 비교했습니다.

LANE C · SELECTIVE RECOVERY

Kafka 상태와 payload로 선택 복구 범위를 정했습니다
Kafka 9개 이벤트를 6개 상태로 구분하고 PostgreSQL 상태와 JSONB payload를 보존했습니다. 8개 복구 경로에서는 완료되지 않은 업무 가운데 포인트 차감·회수·만료 알림만 골라 재실행했습니다.

배치 재시작 · 동시성 검증 · Kafka 선택 복구

서로 연결된 실행 순서가 아니라 각각 독립적으로 확인하는 세 운영 경계입니다.



발표·글·코드로 남긴 공개 기록

기술 글 156편과 YouthCon 발표, Spring 오픈소스 개선 제안, 개인 프로젝트를 한곳에 모았습니다.

[YouthCon 2025 발표](#)

[기술 글 156편 · 2026.07 기준](#)

[Spring 오픈소스 · 개선 제안 3건 반영](#)

[OpenSearch 2.19 한글 플러그인 · Kotlin 이식](#)

[약 7만 건의 삭제 상태 불일치](#)

[JPA 오류 재현과 약 200만 건 보정](#)