PORTFOLIO · DATA ENGINEERING

최성조

데이터 엔지니어 · 데이터 엔지니어링 팀 리드

데이터를 옮기고 쌓고 꺼내 쓰는 일을 합니다. 파이프라인은 한 번 잘 도는 것보다 실패했을 때 어디서부터 다시 돌릴 수 있는지가 중요하다고 생각하고, 그 기준으로 설계합니다.

18년 9개월
개발 경력 · 웹 → 백엔드 → 데이터

팀 리드
데이터 엔지니어 팀

AWS 기술 블로그
검색 시스템 구축·운영 기고

Spark · Airflow · MWAA · Kafka · MSK · BigQuery · Delta Lake · OpenSearch · Terraform · AWS · GCP · Java · Kotlin · Scala · Python

---

CONTACT

Email	csj4032@gmail.com
GitHub	github.com/csj4032
LinkedIn	linkedin.com/in/성조-최
Blog	csj4032.github.io
기고	AWS 기술 블로그 — 검색 시스템 구축과 운영
Location	Seoul, Korea

CONTENTS

01 요약과 강점

SUMMARY

18년 9개월째 개발을 하고 있습니다. 웹 개발로 시작해 백엔드를 거쳐, 최근 몇 년은 데이터 엔지니어링에 집중하고 있습니다. Spark로 대용량을 처리하고 Airflow로 워크플로를 엮으며, Delta Lake 기반의 레이크하우스 구조를 들여다보고 있습니다. JVM(Java·Kotlin·Scala)과 Python을 오가며 파이프라인을 떠받치는 백엔드도 함께 만듭니다.

경력 흐름

START	NEXT	THEN	NOW
웹 개발	**백엔드 개발**	**추천 · 검색**	**데이터 엔지니어링 · 팀 리드**

강점

다시 돌릴 수 있는 파이프라인

정상 동작보다 실패 지점부터의 재처리를 먼저 설계합니다. 어디서부터 다시 돌릴 수 있는지가 파이프라인의 실제 수명을 정한다고 봅니다.

멀티클라우드 데이터 이동

AWS의 실시간·IoT 데이터를 GCP BigQuery로 모으는 하이브리드 구조를 설계했습니다. 두 클라우드를 잇는 네트워크와 OIDC 인증까지 직접 다뤘습니다.

파이프라인과 그 바닥까지

데브옵스 공석 기간에 데이터 플랫폼 인프라를 직접 맡았습니다. MWAA·EMR·MSK를 세팅하고 Terraform으로 관리하며, 위에서 쓰는 것이 아래에서 어떻게 서 있는지 압니다.

품질을 게이트로 강제

Deequ와 Great Expectations로 산출물에 검증을 걸고, 코드는 헥사고날 구조의 의존성 방향과 규약을 테스트로 막아 둡니다. 사람의 주의력에 기대지 않습니다.

저장소를 역할로 나눠 쓰는 관점

저장소는 하나로 뭉치기보다 질문의 성격에 맞게 나눠 씁니다. 정합성이 필요한 정본과 관계 탐색, 본문 검색은 요구가 서로 다르다고 보기 때문입니다.

역할	선택	이유
정본 · 정합성	MySQL · Aurora	트랜잭션이 필요한 원장 데이터
관계 탐색	Neo4j	개체 사이 연결을 질의로 따라가야 할 때
본문 검색 · 임베딩	Elasticsearch · OpenSearch	텍스트 검색과 벡터 검색을 한자리에서
로그성 문서	MongoDB	스키마가 자주 바뀌는 반정형 기록
캐시	Redis · Couchbase	반복 조회 비용 절감
분석 · 웨어하우스	BigQuery · Redshift	대량 스캔과 집계

> **설계 기준** — 파이프라인은 정상 경로보다 실패 경로를 먼저 봅니다. 어느 단계에서 멈췄는지, 어디서부터 다시 돌릴 수 있는지, 다시 돌렸을 때 같은 결과가 나오는지를 기준으로 단계를 자릅니다. 적재를 메달리온 계층으로 나눈 것도, 산출물마다 검증을 게이트로 건 것도 결국 재처리 범위를 좁히기 위한 선택이었습니다.

01 멀티클라우드 데이터 플랫폼

AWS에서 발생하는 실시간·IoT 데이터를 GCP BigQuery로 모으는 하이브리드 구조 설계

맥락 데이터는 AWS에서 발생하는데 분석 수요는 BigQuery에 있었습니다. 두 클라우드를 하나의 플랫폼처럼 쓰게 만드는 것이 과제였습니다.

한 일

- AWS ↔ GCP 하이브리드 데이터 플랫폼 아키텍처 설계
- 메달리온 4계층으로 적재 단계를 나누고 계층별 책임을 분리
- 기존 웨어하우스 마이그레이션 수행
- CDC 파이프라인을 붙여 원본 변경이 분석 계층까지 따라오도록 구성

결과 실시간 데이터와 원장 변경분이 하나의 분석 계층에 모였고, 계층 분리로 재처리 범위를 단계 단위로 좁힐 수 있게 됐습니다.

AWS · GCP · BigQuery · 메달리온 아키텍처 · CDC · DWH 마이그레이션

02 Airflow 워크플로 이관

EC2에서 자체 운영하던 Airflow를 매니지드 환경(MWAA)으로 이관하고 배포 체계를 정립

맥락 EC2 위에서 직접 띄워 쓰던 Airflow는 운영 부담이 컸고, DAG 배포가 체계 없이 이뤄지고 있었습니다.

한 일

- EC2 자체 운영 Airflow를 매니지드 환경으로 이관
- DAG 배포를 위한 CI/CD 파이프라인 구성
- 형상관리 체계를 함께 세워 변경 이력이 남도록 정리

결과 인프라 운영 부담을 매니지드로 넘기고, DAG 변경이 검토와 배포 절차를 거치도록 만들었습니다.

Airflow · MWAA · CI/CD · 형상관리

> **기고** — AWS 기술 블로그에 「AWS 서비스를 활용한 검색 시스템 구축과 운영」을 기고했습니다. 검색 시스템을 만드는 과정과 운영에서 마주친 문제를 정리한 글입니다.
> aws.amazon.com/ko/blogs/tech/building-and-operating-search-system-using-aws-services

03 실시간 스트리밍 파이프라인

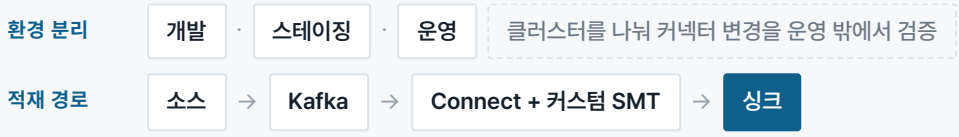
Kafka 클러스터를 환경별로 구축하고 Connect·커스텀 SMT로 적재 경로 구성

한 일

- Kafka 클러스터를 개발·스테이징·운영 세 환경으로 나눠 구축
- Kafka Connect로 적재 경로를 구성하고, 표준 커넥터로 부족한 변환은 **커스텀 SMT**를 만들어 처리
- MSK Connect를 통해 BigQuery로 실시간 적재되는 경로 구성

결과 환경 분리로 운영 토픽에 영향을 주지 않고 커넥터 변경을 검증할 수 있게 됐습니다.

스트리밍 구성



표준 커넥터로 처리되지 않는 변환은 커스텀 SMT 를 만들어 파이프라인 안에서 해결했습니다.

Kafka | MSK | Kafka Connect | 커스텀 SMT | BigQuery

04 데이터 품질 검증과 검색·RAG

파이프라인 산출물에 검증 게이트를 걸고, 한국어 검색과 벡터 검색 인덱싱 파이프라인 구축

품질 파이프라인 산출물에 **Deequ**와 **Great Expectations**로 검증을 걸었습니다. 통과하지 못한 산출물이 다음 단계로 넘어가지 않도록 게이트로 두는 방식입니다.

검색 OpenSearch에 **한국어 형태소 분석**과 **벡터 검색**을 얹고 인덱싱 파이프라인을 만들었습니다. 키워드 검색과 임베딩 기반 검색을 함께 쓸 수 있는 구성입니다.

Deequ | Great Expectations | OpenSearch | 형태소 분석 | Vector Search | RAG

05 추천과 검색 (이전 경력)

데이터 엔지니어링으로 넘어오기 전, 추천과 검색이 주된 일이었습니다

- **추천** — 협업 필터와 Association Rule 기반 추천 로직 구현
- **데이터 마트** — 상품 데이터 마트 구축
- **행동 로그** — 사용자 행동 로그 ETL 파이프라인 구성
- **웨어하우스** — 데이터브릭스 기반 웨어하우스 구축

Collaborative Filtering | Association Rule | Databricks | ETL | Data Mart

04 인프라 담당과 개인 프로젝트

INFRA & PERSONAL

데브옵스 담당이 공석인 기간 동안 데이터 플랫폼의 인프라를 직접 맡았습니다. 파이프라인을 만드는 일과 그것이 돌아갈 바닥을 까는 일이 분리되지 않는 상황이었는데, 덕분에 위에서 쓰는 것들이 아래에서 어떻게 서 있는지 알게 됐습니다.

영역	담당한 내용
분석·처리 환경	MWAA, EMR Studio, SageMaker 세팅. EMR Spark 실행 이미지를 ECR로 관리
스트리밍 인프라	MSK 클러스터와 MSK Connect 구성. 보안 그룹·서브넷·Secrets·KMS 등 주변 설정 포함
AWS ↔ GCP 연동	MSK Connect로 BigQuery 실시간 적재, Aurora를 Datastream으로 BigQuery에 CDC 복제. 두 클라우드를 잇는 네트워크와 OIDC 인증 설정이 가장 손이 많이 간 부분
데이터베이스 운영	RDS Aurora 파라미터 그룹 관리
설정 관리	기존 Terraform 구성 위에서 필요한 부분을 수정하고 배포 관리. Secrets Manager, IAM 역할·정책, 계정 권한이 주 대상

> 전담은 아니었지만, 데이터 처리를 위한 AWS 인프라가 어떻게 구성되는지 이해하고 있고 필요한 설정은 직접 할 수 있습니다.

개인 프로젝트 — 투자 정보 자동화 플랫폼

SIDE

수집부터 발행까지 한 줄기로 묶은 자동화 플랫폼을 만들고 있습니다. 앞에서 정리한 저장소 분리 관점을 실제로 적용해 본 사례입니다.

수집	저장	분석	발행
공시 · 리포트 · 뉴스 · 시세	**MySQL · Neo4j · ES**	**스크리닝 · 백테스트**	**가중치 보정**

- **정본**은 MySQL에 두고, 종목 사이의 **관계**는 Neo4j에, **본문**은 임베딩해 Elasticsearch에 넣습니다.
- 그 위에 **모멘텀·가치주 스크리닝**과 **백테스트**를 얹었습니다.
- 가중치는 감으로 정하지 않고, 라벨이 쌓인 뒤 **실제 수익률과의 상관**을 보고 조정합니다.

JVM(Java·Kotlin·Scala) · Python | MySQL | Neo4j | Elasticsearch | 임베딩 · 벡터 검색 | 백테스트 | 헥사고날 아키텍처

코드 구조에 대한 원칙

- 구조는 **헥사고날**로 두어 domain → application → infrastructure 방향을 **테스트로 강제**합니다.
- 함수 길이나 파라미터 수 같은 규약도 **게이트로 막아** 둡니다.
- 잘 도는 코드보다 고쳐 쓸 수 있는 코드가 오래 간다고 생각합니다.

05 기술 스택 · 저장소 · 기록

STACK & RECORDS

기술 스택

Data	Spark, Airflow(MWAA), Kafka(MSK), Hadoop, EMR, Databricks, Deequ, Great Expectations, Pandas
Infra	Terraform, IAM, Secrets Manager, MSK Connect, Datastream, ECR, EMR Studio, SageMaker
Storage	BigQuery, MySQL·Aurora, PostgreSQL, Elasticsearch·OpenSearch, Redshift, MongoDB, Neo4j, Redis, InfluxDB
Language	Java, Kotlin, Scala, Python, SQL, JavaScript
Cloud	AWS, GCP
자격	빅데이터분석기사, ADSP, SQLD, DASP, 정보처리기사

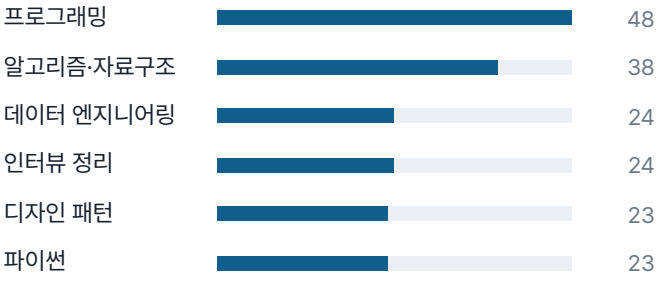
공개 저장소 — github.com/csj4032 · 공개 저장소 18개

저장소	내용	STARS
enjoy-docker	데이터 엔지니어링용 Docker 인프라 스택. MySQL·PostgreSQL·Redis·MongoDB·Neo4j, MinIO, Confluent Kafka, Flink를 한 벌로 구성	—
enjoy-workflow	Airflow 기반 워크플로·ETL 파이프라인. 다중 소스 연동, Spark·Livy 분산 처리, Great Expectations·Deequ 검증, Debezium CDC	—
enjoy-workreduce	AWS EMR Serverless용 Python 패키지. PySpark 데이터 처리와 PyDeequ 품질 검증, PyFlink 기반 Kafka CDC 스트리밍 예제	1
primavera	스프링부트를 이용한 커뮤니티 사이트 개발	58
enjoy-design-pattern	『Java 언어로 배우는 디자인 패턴 입문』 예제 구현	14
enjoy-algorithm	백준 온라인 저지, algospot, codingdojang, codility 알고리즘 풀이	1
csj4032.github.io	기술 블로그 (Jekyll)	—

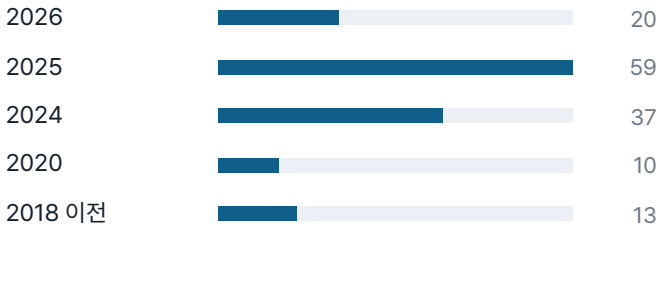
이 밖에 enjoy-spring, enjoy-tomcat, enjoy-java-books, javaProgAlgo 등 학습 저장소가 있고, Apache Spark · Airflow · Delta Lake 저장소를 포크해 코드를 따라 읽고 있습니다.

기술 블로그 — csj4032.github.io · 139편

주제별 글 수



연도별 글 수



Spark의 셔플과 파티션, AQE, Airflow 오케스트레이션, Kafka 스트리밍, BigQuery 데이터 모델링 같은 주제를 질문과 답변 형태로 정리해 두고 있습니다. 알고리즘·자료구조·디자인 패턴 글은 실무에서 다시 익힌 것들의 기록입니다.

> **연락** — csj4032@gmail.com · github.com/csj4032 · csj4032.github.io