데이터 엔지니어 최승호

seungho546@naver.com | GitHub | LinkedIn | Portfolio | Seoul, Korea

소개

데이터가 잘 흐르고 다양하게 활용될 수 있도록 꿈꾸고 실현하는 **5년차 데이터 엔지니어 최승호**입니다.

어떻게 하면 **안정적인 데이터 파이프라인**을 구축할지, 어떻게 하면 **비용 효율적인 데이터 플랫폼**을 구성할지, 어떻게 하면 **데이터 분석에 집중할 수 있는 환경**을 제공할지 지속적으로 고민하고 테스트하며 도입합니다.

데이터를 통해 비즈니스 가치를 창출하고, 조직이 공통된 목표를 향해 나아갈 수 있도록 커뮤니케이션하며 기여하는 것을 목표로 합니다.

경력

Neowiz | Data Engineer

2020.07 - 현재

실시간(CDC) 데이터 파이프라인 구축 및 데이터 웨어하우스 운영을 담당하며 다음과 같은 핵심 성과를 달성했습니다.

- 🔄 **15개 이상 다양한 데이터 소스 통합 및 일 10억 건+ CDC ETL 구축**: 파편화된 데이터를 통합하여 분석 기반을 마련했습니다.
- 📊 **Redshift 멀티클러스터 아키텍처 설계**: 성능 병목을 해결하고 데이터 매시 구조의 초석을 마련했습니다.
- 🏗️ **멀티클라우드(AWS ↔ GCP) 실시간 데이터 파이프라인 구축**: 일 4,000만 건 이상의 데이터를 처리하며 실시간 분석 및 FDS를 지원했습니다.
- 🧊 **Trino와 Iceberg를 활용한 데이터 레이크 아키텍처 설계**: 데이터 접근성을 높이고 DW 부하를 분산시켰습니다.
- ⚡ **자동화 및 모니터링 시스템으로 운영 리소스 90% 절감**: IaC, Grafana, Prefect 등을 활용하여 안정적인 플랫폼 운영을 달성했습니다.
- 💰 **인프라 비용 최적화로 고정비용 20% ($3,000+) 절감**: Graviton 전환, 서버리스 아키텍처 도입, 유휴 리소스 자동 관리를 통해 비용 효율을 극대화했습니다.
- 🤖 **LLM 기반 Text-to-SQL 시스템으로 데이터 추출 요청 40% 감소**: 데이터 민주화를 실현하고 팀의 핵심 업무 집중도를 높였습니다.

보유 기술

- **Specialties**: Real-time(CDC) Data Pipeline, Multi-cloud Architecture, Cost Optimization, Data Governance
- **Cloud Platforms**: AWS, GCP
- **Data Engineering**: Prefect, Apache Kafka, Trino
- **Data Warehouse**: Redshift, BigQuery, Snowflake
- **Databases**: MySQL, PostgreSQL, DynamoDB, ElasticSearch, Redis
- **Programming**: Python, SQL, Java
- **Infrastructure**: Terraform, Docker, ECS, Grafana

- **AI/ML**: LangChain, Langfuse, RedshiftML, SageMaker, OpenAI GPT

주요 프로젝트

1. AWS 멀티 클러스터 아키텍처 도입 (GAMES ON AWS 2024 발표)

- **문제**: 단일 Redshift 클러스터의 성능 병목 및 확장성 한계에 직면.
- **해결**: Redshift Serverless와 Data Sharing을 도입하여 워크로드별(ELT, BI, AI/ML) 클러스터로 부하를 분산하고, Zero-ETL을 통해 CDC 처리 병목을 해소했습니다. 180TB에 달하는 대규모 클러스터 암호화 전환 시, 사전 데이터 최적화 및 엔드포인트 스위칭 방식으로 다운타임을 30분 이내로 최소화했습니다.
- **성과**: 쿼리 성능 50% 향상, 운영 안정성 강화, 데이터 매시 구조의 기반 마련.

2. Multi-Cloud Real-time Data Pipeline (AWS ↔ GCP)

- **문제**: 글로벌 서비스의 일 배치 시스템을 준실시간으로 전환해야 하는 요구사항 발생.
- **해결**: AWS DMS, Lambda, SQS를 활용하여 RDS(Aurora) 데이터를 GCP BigQuery로 이전하는 파이프라인을 구축했습니다. SQS를 통한 재처리 구조, BigQuery Job ID를 활용한 중복 방지, CD 기반의 테이블 관리 자동화 등을 통해 안정성을 확보했습니다.
- **성과**: 평균 1-2분의 지연 시간을 갖는 준실시간 파이프라인을 구축하여 실시간 대시보드 및 FDS 지원 기반을 마련하고, ETL 관리 리소스를 대폭 절감했습니다.

3. Trino on ECS 기반 DataLake 플랫폼

- **문제**: 데이터 레이크 부재로 인한 데이터 활용성 저하 및 DW 의존성 심화.
- **해결**: AWS ECS 환경에 Trino를 직접 배포하여 Redshift, Aurora, S3 등 다양한 소스를 통합 쿼리할 수 있는 Federated Query 플랫폼을 구축했습니다. ECS Service Connect, Auto Scaling, Parameter Store를 활용하여 안정성과 운영 편의성을 확보하고, S3 Lifecycle 정책으로 스토리지 비용을 40% 절감했습니다.
- **성과**: 분석가의 원본 데이터 탐색 효율 증대, DW 부하 감소, Lakehouse 아키텍처의 기술적 토대 마련.

4. 스트리밍 데이터 수집 플랫폼 구축

- **문제**: 비정형/반정형(클라이언트 로그, DynamoDB) 실시간 데이터 처리 인프라 부재.
- **해결**: Amazon MSK를 중앙 데이터 허브로 구축하고, MSK Connect를 통해 Snowflake, Redshift 등 다양한 타겟에 Code-less로 데이터를 연동했습니다. DynamoDB Streams 데이터는 멱등성을 보장하는 UPSERT 쿼리로 안정적으로 처리하고, Terraform 템플릿을 통해 Cross-Account 접근 제어를 구현했습니다.
- **성과**: 이벤트 기반의 심층 분석 환경 마련, 데이터 사일로 해소, 데이터 수집 파이프라인 표준화 및 안정성 증대.

5. LLM 기반 Text-to-SQL 시스템

- **문제**: 반복적인 Ad-hoc 데이터 추출 요청으로 인한 데이터 팀의 리소스 분산.
- **해결**: LangChain과 OpenAI GPT를 활용하여 자연어 질문을 SQL로 변환하는 시스템을 구축했습니다. Hybrid Search, Re-ranker, Few-shot 프롬프팅으로 정확도를 높이고, Langfuse를 도입하여 LLM의 작동을 추적하고 개선했습니다.
- **성과**: 데이터 추출 요청 40% 감소, 데이터 민주화 및 전사적 데이터 활용 문화 확산, 엔지니어링 생산성 증대.