데이터 엔지니어

최재영

2017 ~ 2022 Java 기반 백엔드 개발

2022 ~ Python, SQL 기반 데이터 엔지니어링

과제 1. 데이터 파이프라인 신규 구축

개요

기간: 2022년 11월 ~ 2023년 2월

인원: 데이터 엔지니어 1명, 인프라 1명

목적: 내부 비즈니스 운영 영역의 디지털 전환(DX)

배경

- 전통적인 운영 업무 프로세스에 대한 RPA 구현에 대한 제안
  (운영 데이터 엑셀 추출 → 엑셀 차트 구현 → PDF 출력 프로세스 개선)
- 운영 Application 성능 저하 이슈

과정

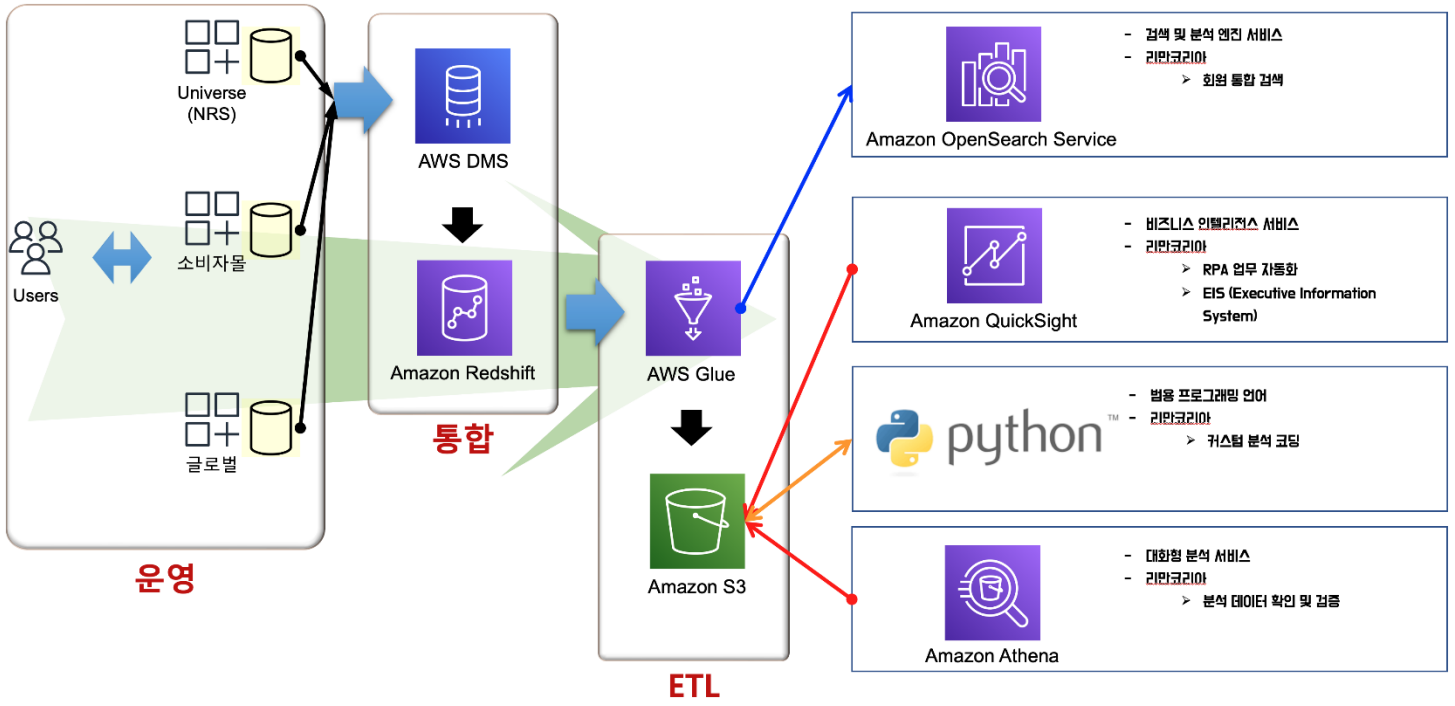
1) 현업 요구사항 수집 (리포트 장표 수집)
2) CDC 방식으로 데이터 수집 설정 (AWS DMS, Redshift)
3) 1차, 2차 데이터 ETL Job 작성 (AWS Glue)
4) BI Service 설정 (Quicksight, S3, Athena)

리뷰

- 운영 Application에 큰 영향을 미치지 않으면서 데이터를 추출하고 가공할 수 있는 환경 구성
- BI 서비스를 통해 다양한 데이터 표현 방식 제공
- RDBMS에 한정되지 않고 다양한 원천 데이터를 수집할 수 있는 환경 구성이 추가로 필요

나의 역할 : 데이터 엔지니어

- 현업 요구사항을 분석하고 데이터 수집을 위한 파이프라인을 설계.
- CDC 방식을 활용하기 위해 AWS DMS 설정.
- AWS Glue를 사용하여 ETL 작업을 개발. (Python)
- Quicksight, S3, Athena를 통해 BI 서비스를 설정.



과제 2. 데이터 분석 플랫폼(DAP) 고도화

개요

기간: 2023년 6월 ~ 2023년 8월

인원: 데이터 엔지니어 2명, 인프라 1명

목적: 기존 DAP의 이슈 해결 및 아키텍처 고도화

배경

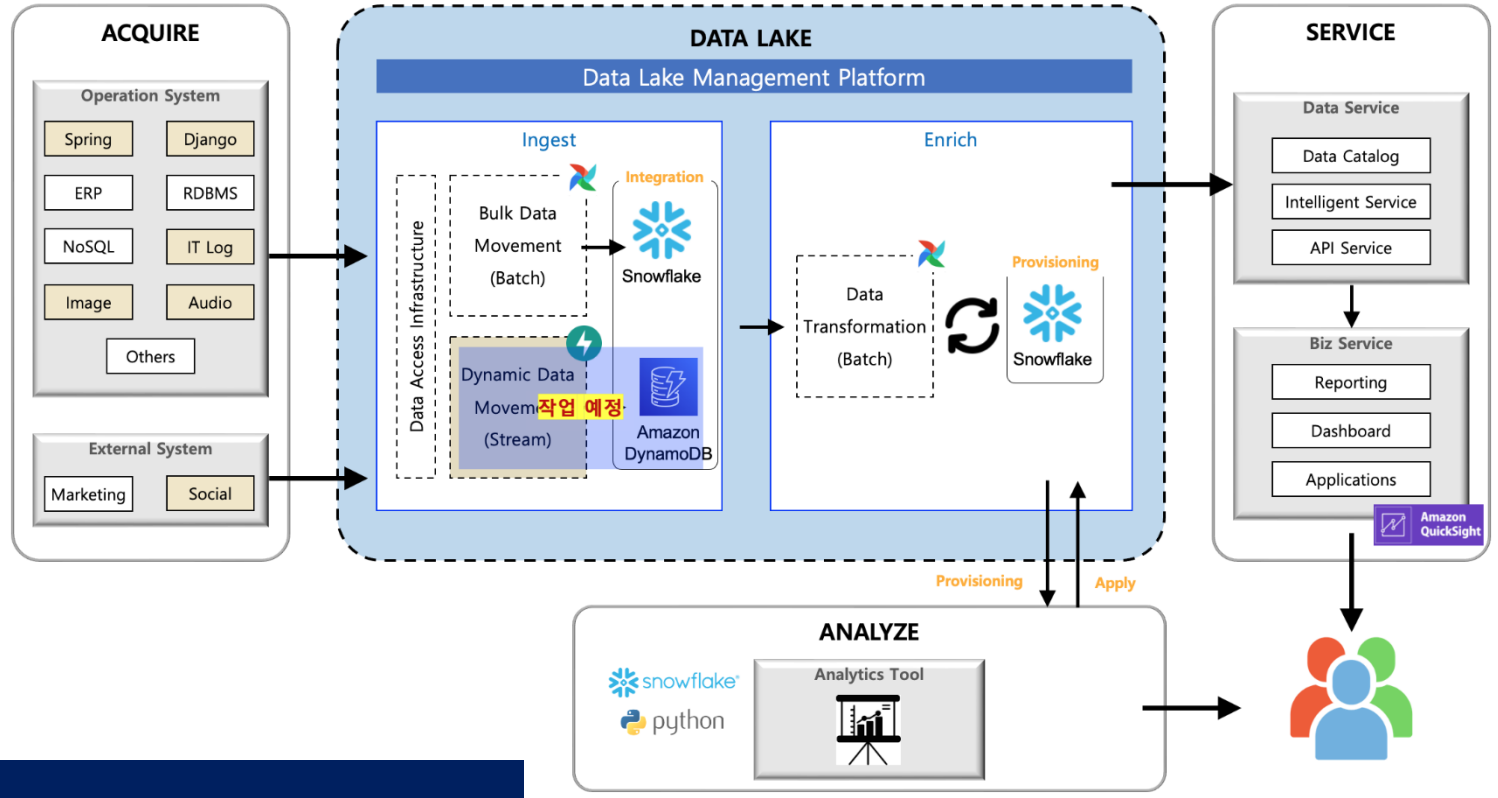
- AWS DMS의 CDC 작업 비용이 예상보다 크게 발생하여 Redshift를 최고 사양으로 변경하는 이슈 발생 (비용 부담)
- AWS Glue의 데이터 변환 작업에 선후관계를 관리하는 정책이 적용되어 있지 않음
- RDBMS 이외의 다양한 데이터 소스를 지원하지 않음

과정

1) Airflow 및 CI/CD 환경 구축 (EC2, Docker, Jenkins)
2) Snowflake 인프라 환경 구축 (PrivateLink)
3) 원천 데이터 수집 파이프라인 구성 (L0, Level 0)
4) 1차, 2차 가공 데이터 파이프라인 구성 (L1, L2)
5) BI Service 연동 및 기존 데이터 세트 교체 작업

리뷰

- 데이터 수집 방식 변경으로 데이터의 성격에 맞는 방식을 선택할 수 있게 되었습니다.
  [CDC → Batch / Stream]
- 워크플로우 관리 도구의 적용으로 ETL 작업의 선후관계를 조정할 수 있게 되었습니다.



나의 역할 : 데이터 엔지니어

- Airflow를 활용하여 워크플로우 관리 도구를 구축. (Python)
- CI/CD 환경을 구축하여 소스 코드의 자동 빌드, 테스트, 배포를 환경 구축.
- Snowflake 인프라 환경을 구축 및 구성.
- 원천 데이터 소스를 수집하는 파이프라인 구성 (L0, Level 0).
- BI 서비스와의 연동을 설정하고 기존 데이터 세트를 교체하는 작업을 수행.

과제 3. Snowflake RBAC 개선 작업

개요

기간: 2023년 12월 ~ 2024년 1월

인원: 데이터 엔지니어 1명

목적: 역할 기반 액세스 제어(RBAC) 강화를 통해 사용자의 접근을 단순화

배경

- 무질서한 Role 정책으로 Object 소유권 분쟁 이슈 발생
- Snowflake Guide 권장사항
- Role 기반 비용 모니터링에 대한 제안

과정

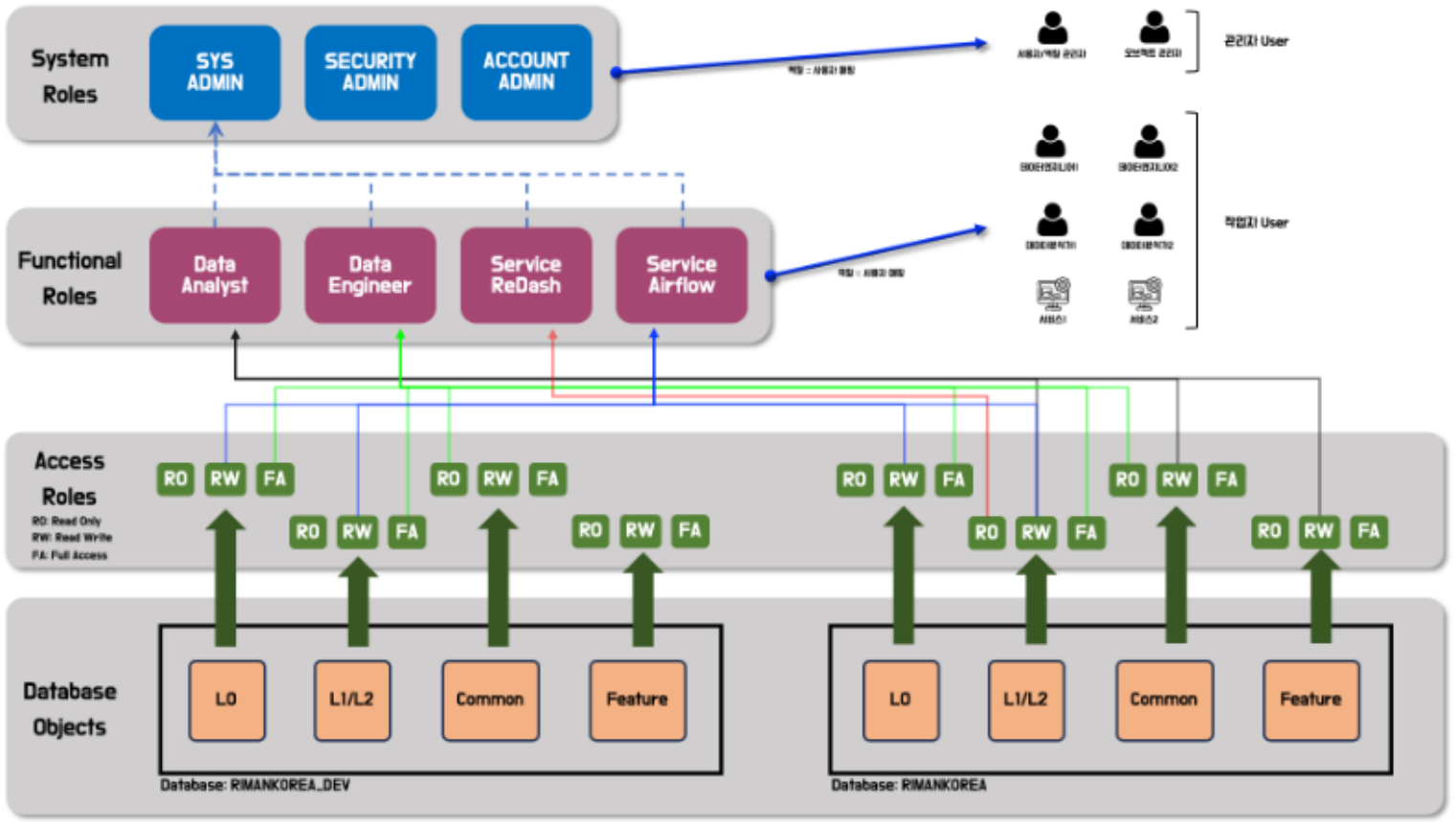
1) Role 생성 (사용자 기반: Functional Roles / 오브젝트 기반: Access Roles)
2) Role간 상속관계 설정 및 사용자에게 역할 부여
3) 사용자 테스트 및 운영환경 적용

리뷰

- 오브젝트에 대한 소유권이 명확해짐.
- 효율적인 비용 모니터링이 가능해짐.

나의 역할 : 데이터 엔지니어

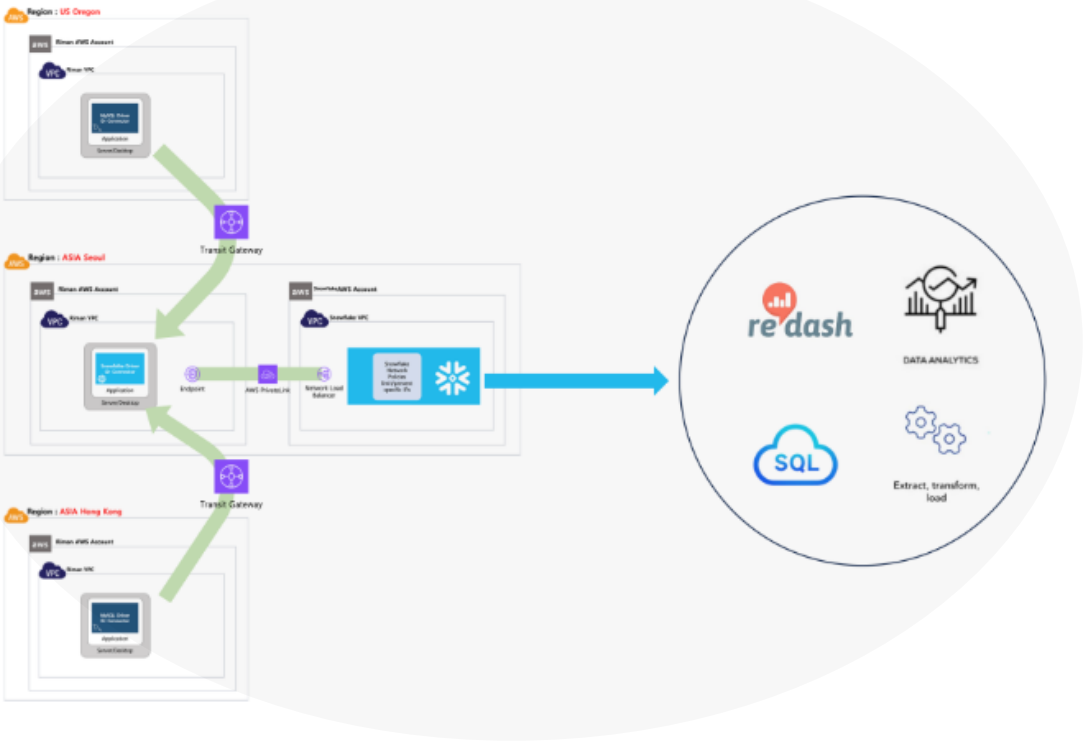
- Snowflake RBAC 개선 작업에서 데이터 엔지니어로 참여.
- 역할 기반 액세스 제어(RBAC)를 강화하기 위해 Functional Roles와 Access Roles를 사용하여 Role을 생성.
- Role간 상속관계를 설정하고 사용자에게 역할을 부여.
- 역할 기반 액세스 제어(RBAC)의 효과를 확인하기 위해 사용자 테스트와 운영환경에 적용.



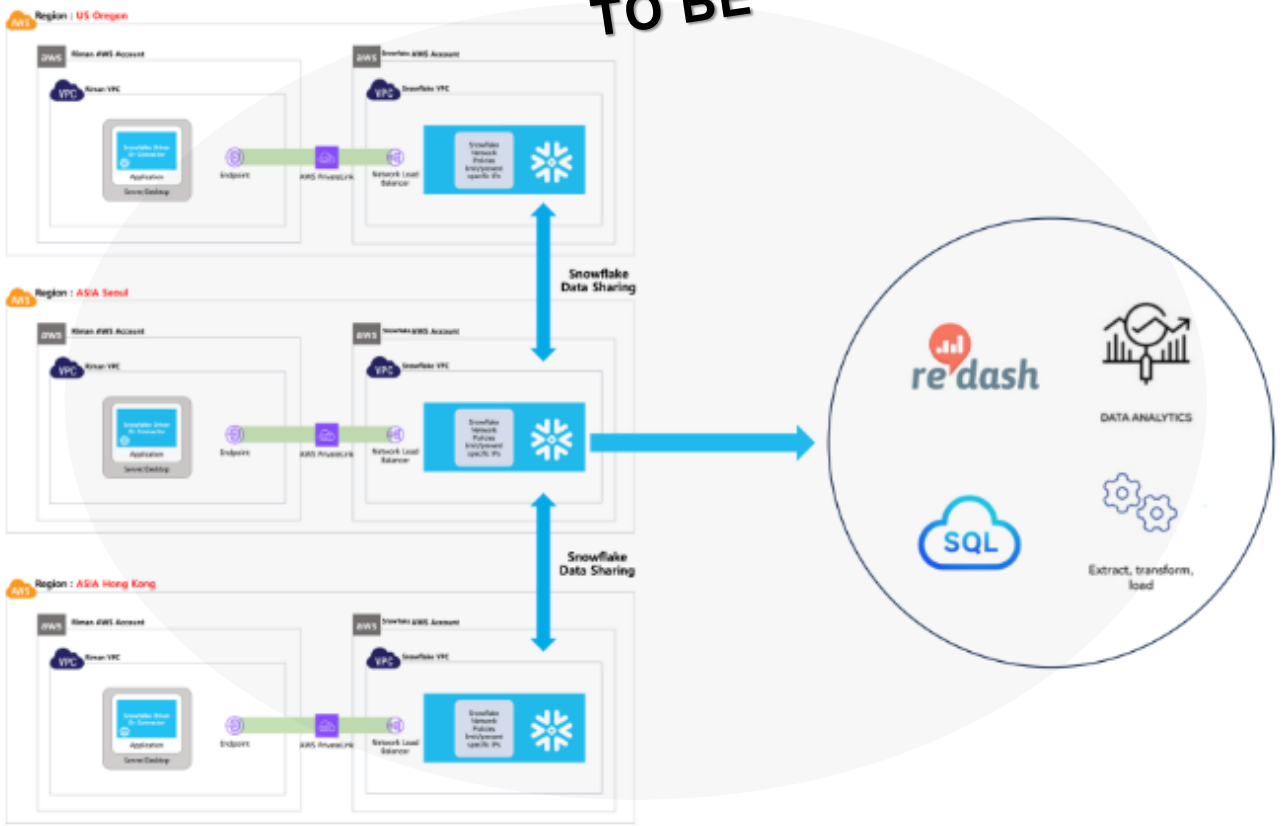
과제 4. Snowflake 다국가 운영 환경구축 (Multi Region)

개요	배경	과정	리뷰	나의 역할 : 데이터 엔지니어
기간: 2024년 1월 ~ 2024년 2월 인원: 데이터 엔지니어 1명, 인프라 1명 목적: 국내 뿐만 아니라 해외(Hongkong, USA) 데이터 통합	- 다국가 사용자 수요 증가 - 비즈니스 요구사항 - 데이터 센터 지리적 분산 - 비용 최적화	1. Hongkong, USA PrivateLink 인프라 구성 2. 각 국가별 데이터 파이프라인 구축 (L0, Level 0) 3. Snowflake Data Sharing 설정	- 단일 리전에 비해 운영비용 40% 절감. (Data Transfer, Compute) - 서울리전의 데이터 수집 처리 과부하 해소.	- Hongkong, USA PrivateLink 인프라를 구성. - 각 국가별 데이터 파이프라인을 구축. - Snowflake Data Sharing을 설정.

AS IS



TO BE



과제 5. 분산 스트리밍 플랫폼 (Apache Kafka) 도입 PoC

개요

기간: 2024년 3월 (10일간)

인원: 데이터 엔지니어 1명

목적: 실시간 분산 스트리밍 플랫폼을 도입하여 데이터 엔지니어링 및 애플리케이션 개발에 활용 가능한 수단을 구축하는 것

배경

- 비즈니스 데이터 마이닝 업무에서 실시간 처리에 대한 요구가 증가하고 있음
- 현재 데이터 수집 및 처리를 Airflow를 통해 진행 중이나, 이는 워크플로우 관리를 위한 목적과는 다른 작업을 수행하고 있어 리소스 및 성능 문제가 발생하고 있음.
- CDC(Change Data Capture) 도구를 AWS DMS에 종속적으로 운영 중이며, 확장성에 제약이 있음

과정

1) Source와 Target 사전 구성

   Source DB (MySQL) 및 Target DB (Snowflake)에 대한 필요한 사전 설정

2) 카프카 클러스터 구축 (AWS EC2 3대)

3) 카프카 커넥터 설치 및구성 (Debezium connector, Snowflake connector)

4) Kafka 데이터 적재 이후 Snowflake 처리방안 적용

   Stream & Task / Dynamic Table

리뷰

- Apache Kafka를 활용한 분산 스트리밍 플랫폼 도입을 완료
- MySQL과 Snowflake와의 CDC 연동 준비를 완료하여 시시간 데이터 처리가 활용 가능
- 서비스 애플리케이션 개발 영역에서 Kafka를 효과적으로 활용 가능

나의 역할 : 데이터 엔지니어

- EC2 3대를 이용하여 카프카 클러스터 구축
- 카프카 커넥터 설정 및 연동
- Snowflake Stream, Task, Dynamic Table을 이용한 데이터 노멀라이징 작업