AWS 서비스를 활용한 검색시스템 구축 및 운영사례

신누리

Index

OpenSearch

1. 구축 배경

이커머스 플랫폼의 검색 시스템 개선 필요성

OpenSearch

이커머스 시스템의 검색 시스템 개선 필요성

- **기존 방식**
  - 타사의 SaaS 검색 솔루션 사용
  - 상품 데이터를 JSON 파일로 추출하여 연계
  - REST API를 통해 검색 질의 수행
- **한계점**
  - **비즈니스 변화 대응 어려움**
    - 비즈니스 정책 변경 시 즉각 반영 불가능
  - **확장성 부족**
    - 새로운 서비스 추가 시 도메인 반영 어려움
  - **느린 색인 반영**
    - 하루 1회 전체 색인, 30분 단위 증분 색인으로 실시간 반영 한계
- **개선 방향**
  - 내부 검색시스템을 구축하여 유연하고 효율적인 운영 가능하도록 개선

OpenSearch

2. 검색 시스템 개요

전체 색인과 증분 색인

OpenSearch

전체색인

전체 색인이란 검색 시스템에서 모든 데이터를 새롭게 추출하고 변환(ETL)하여 검색 인덱스를 재구성하는 과정

- **색인 성능의 핵심 요소**
  - 데이터 추출 및 변환(ETL) 과정이 가장 중요한 과제
  - 클러스터 확장 및 쿼리 최적화를 통한 부차적인 성능 개선 가능
- **기술적 목표**
  - **유연한 스케줄링 및 스크립트 관리** → 색인 프로세스의 효율적 운영
  - **데이터 병렬 처리** → 대용량 데이터의 빠른 추출 및 변환
  - **각 ETL 단계의 명확한 책임 분리** → 유지보수 및 확장성 강화

증분색인

증분 색인이란 변경된 데이터(추가, 수정, 삭제된 항목)만을 선별하여 검색 인덱스에 반영하는 과정

- **색인 성능의 핵심요소**
  - **변경된 데이터만 색인** → 불필요한 데이터 처리 최소화
  - **빠른 데이터 반영** → 검색 결과의 신속한 업데이트 가능
  - **색인 성능 최적화** → 리소스 사용량 절감 및 처리 속도 향상
- **기술적 목표 개선 목표**
  - **관리 복잡성 최소화** → 색인 자동화 및 운영 효율화
  - **데이터 색인 속도 극대화** → 준 실시간(Near real-time) 반영을 위한 최적화

3. 전체 색인 파이프라인

데이터 아키텍처 및 시퀀스

OpenSearch

활용되는 **AWS** 서비스

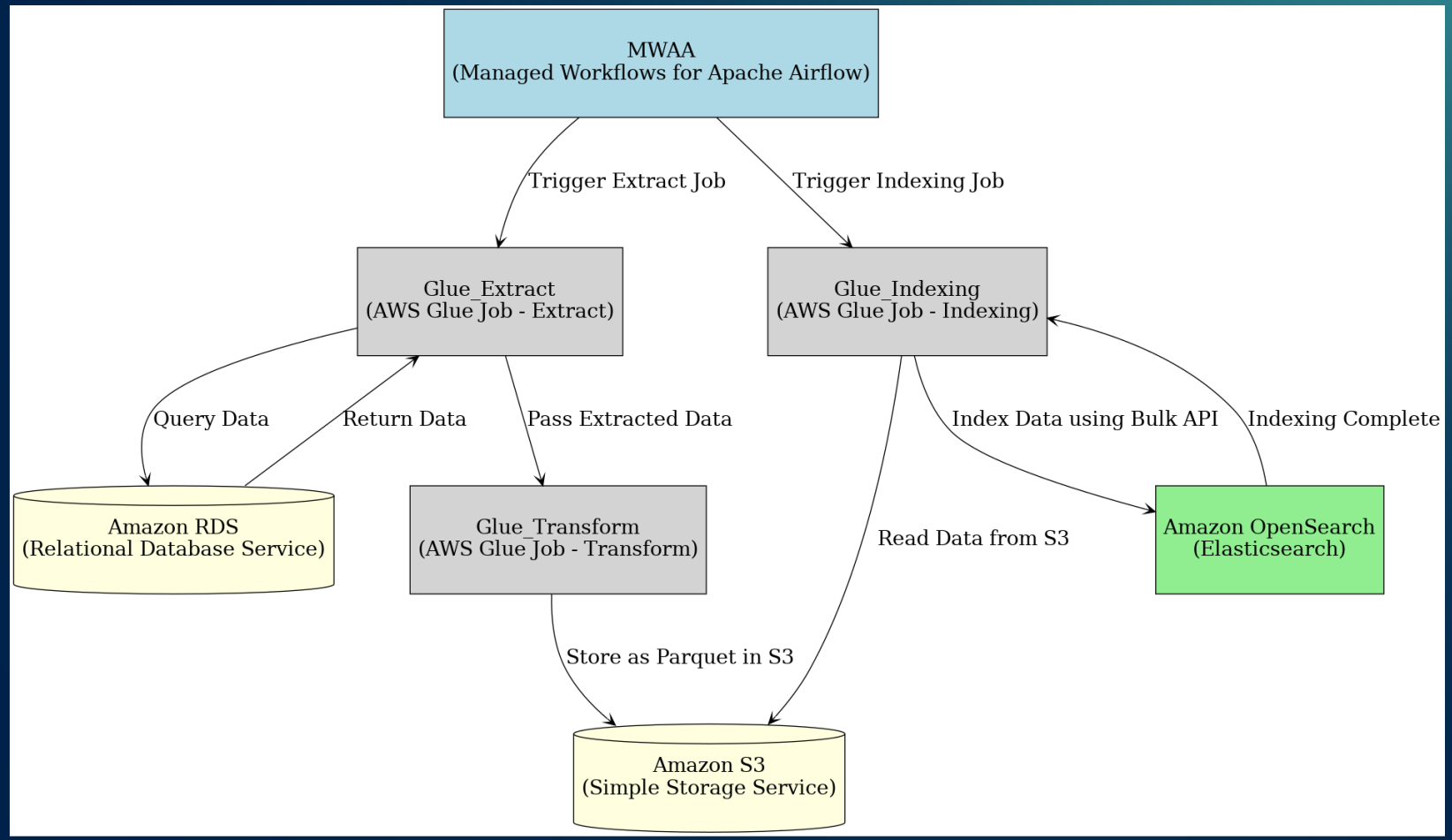
Service	Description
Amazon Managed Workflows for Apache Airflow (MWAA)	Apache Airflow 기반 워크플로 오케스트레이션을 위한 관리형 서비스. 운영 부담 없이 확장성과 보안성 제공.
AWS Glue	데이터 탐색, 준비, 이동 및 통합을 지원하는 서버리스 데이터 통합 서비스.
Amazon Simple Storage Service(Amazon S3)	높은 확장성과 가용성을 제공하는 객체 스토리지 서비스.
Amazon Opensearch Service	실시간 검색, 모니터링 및 분석을 지원하는 검색 및 로그 분석 서비스.

전체 색인 시퀀스

데이터 추출 : Glue Extract

데이터 변환 : Glue Transform

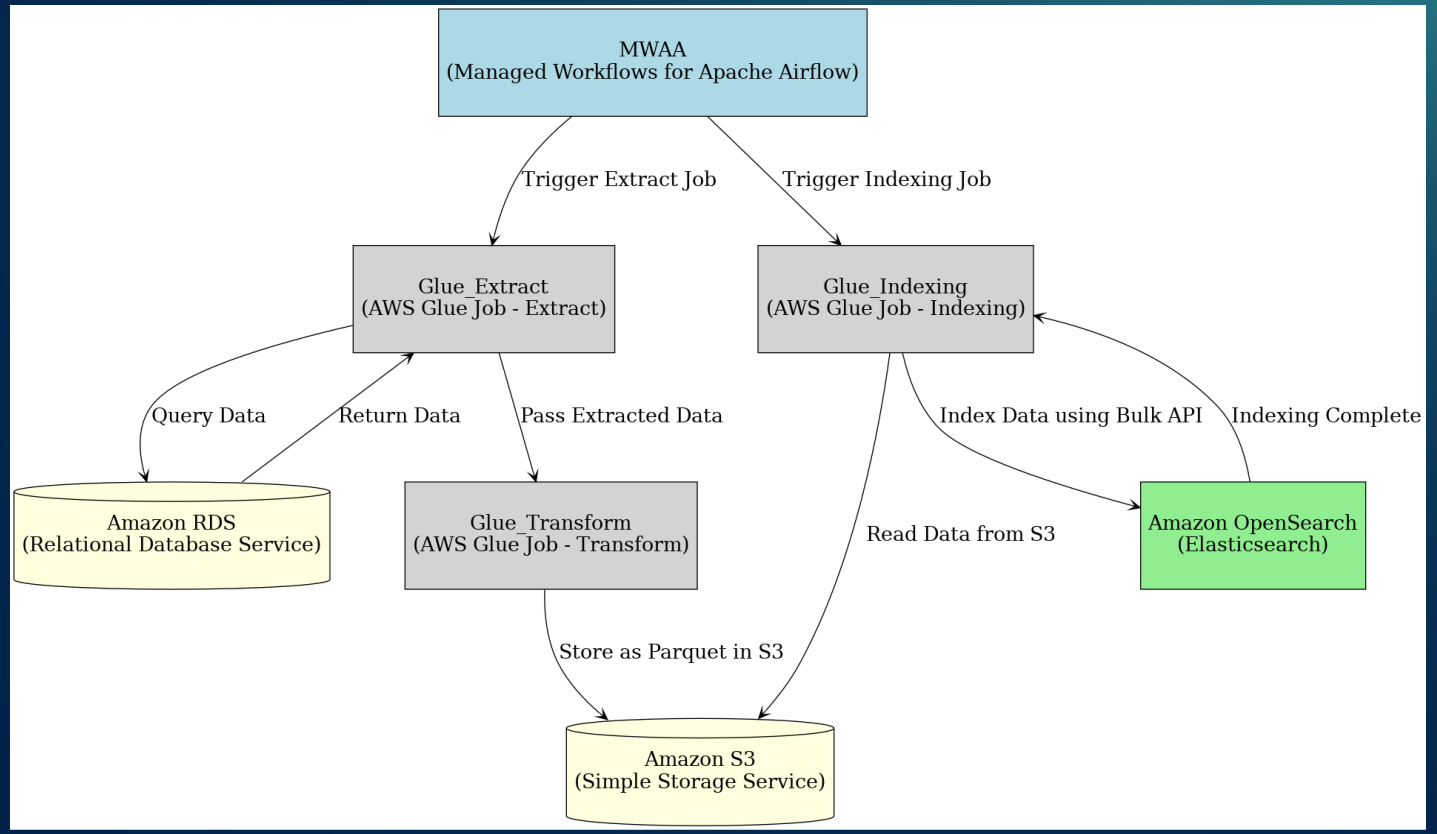
데이터 색인 : Glue Indexing



OpenSearch

데이터 추출

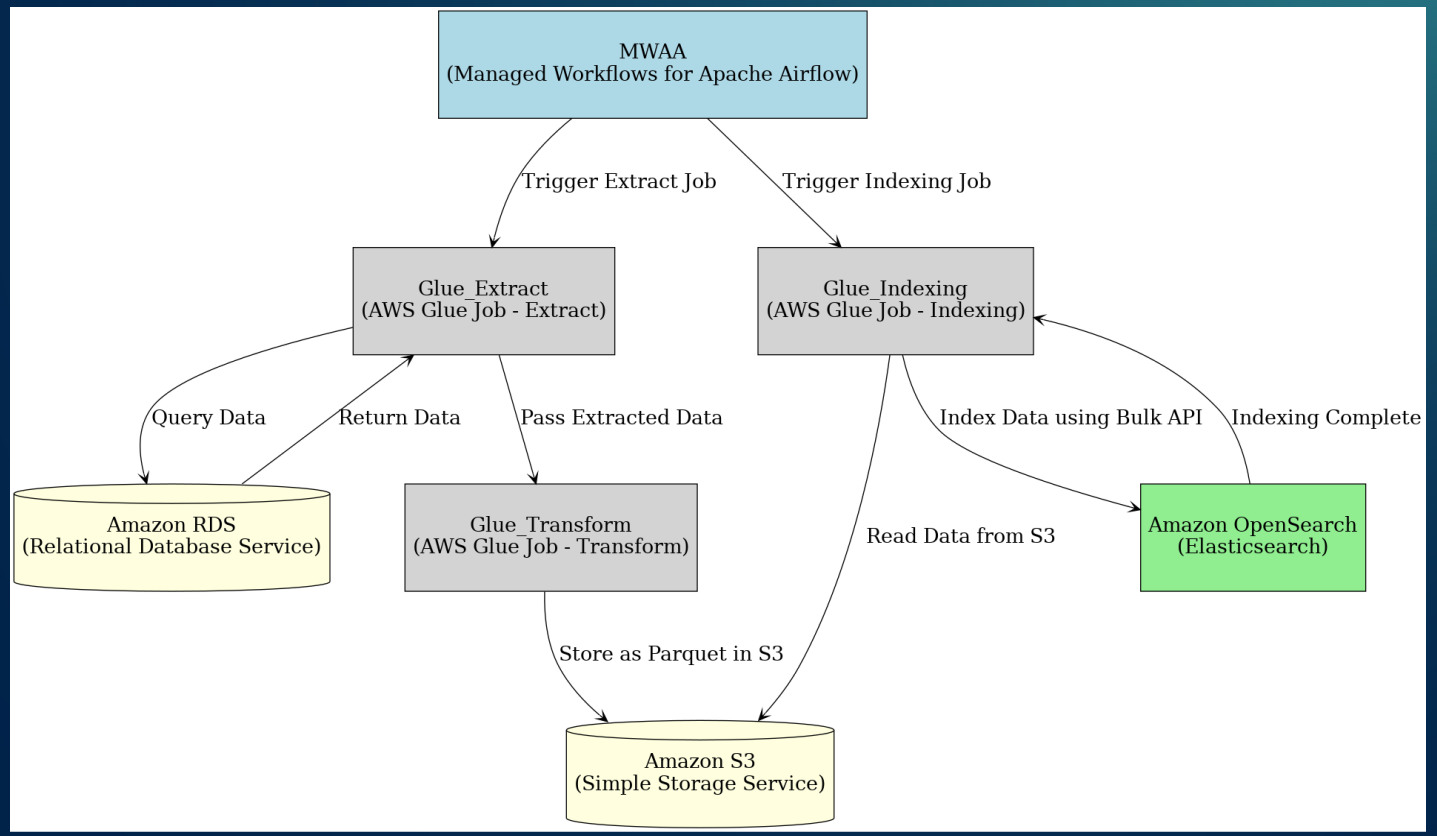
1. MWAA
   - 설정된 스케줄에 따라 데이터 추출을 위한 Glue Job 실행
2. Glue
   - RDS에 저장된 데이터를 AWS Glue의 Apache Spark 기반 스크립트를 사용하여 추출



OpenSearch

데이터 변환

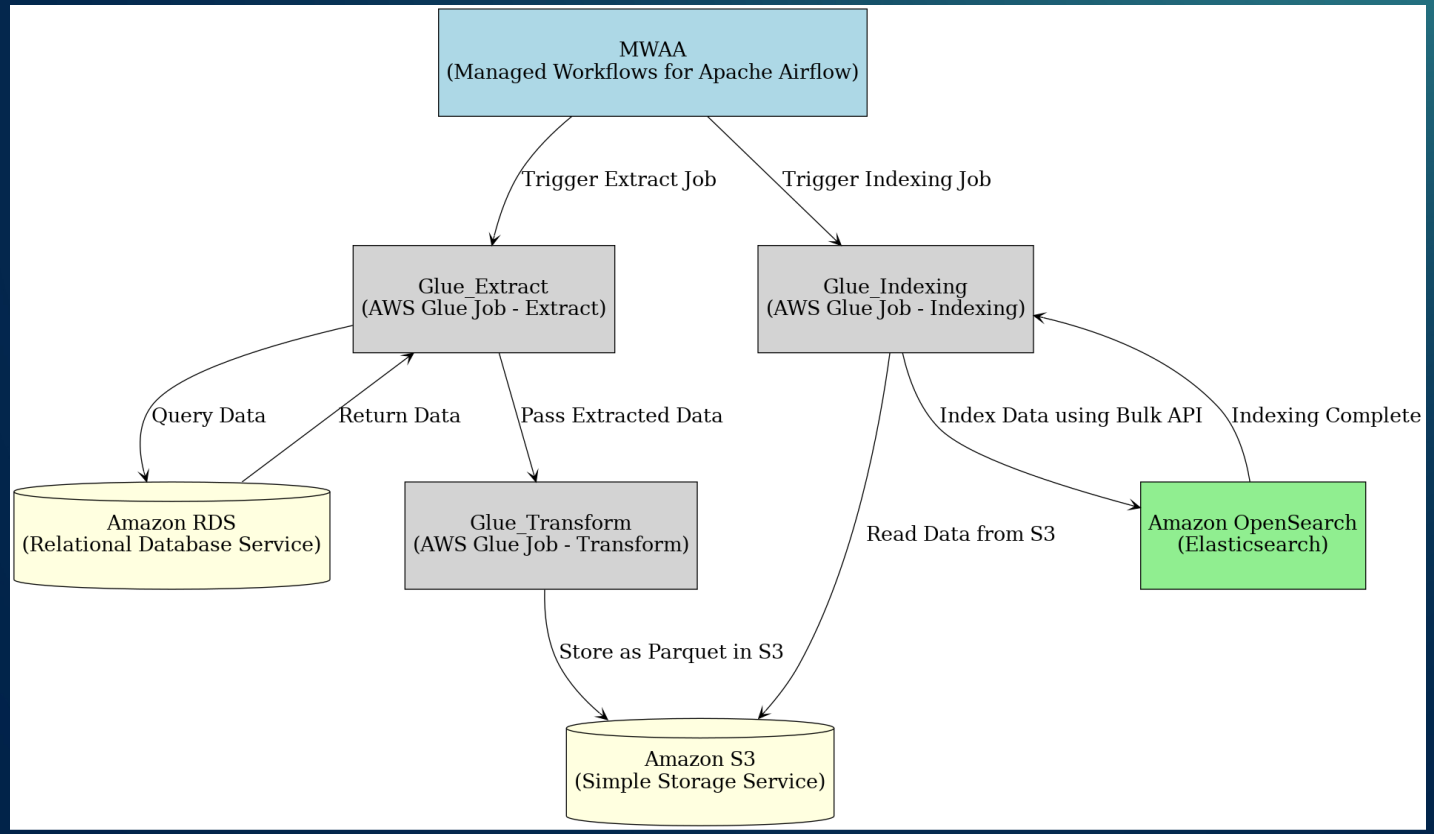
1. Glue
   - 추출된 데이터를 바탕으로 비즈니스 로직 적용 후 Parquet 파일로 변환
2. S3
   - Parquet 포맷의 파일을 S3 버킷에 저장



OpenSearch

데이터 색인

1. MWAA
   - 데이터 추출 Glue Job 종료 후 트리거가 활성화되어 데이터 색인을 위한 Glue Job이 실행
2. Glue
   - S3 버킷에 저장된 데이터를 읽어 OpenSearch의 Bulk API를 통해 인덱싱 작업



OpenSearch

주요 이점

- **스케줄링 및 스크립트 변경 용이성**
  - MWAA를 활용해 워크플로를 효율적으로 관리하고, 스케줄링 및 트리거 설정 간편하게 구성 가능
  - DAG 작업 이력을 콘솔에서 쉽게 확인하여 검색 파이프라인 관리 최적화
- **수행 이력 및 로그 적재**
  - Glue를 통해 ETL 작업의 수행 이력을 추적하고, 로그를 효과적으로 관리 가능
  - 서버리스 환경에서 안정적인 데이터 처리 지원
- **자동화된 데이터 파이프라인 구축**
  - MWAA와 Glue를 활용해 전체 프로세스를 자동화하여 인적 오류를 최소화
  - 일관된 데이터 처리를 보장하고 운영 부담 감소
- **데이터 병렬 처리 최적화**
  - Spark 기반 병렬 처리로 대규모 데이터 분석 성능 극대화
  - 코어 수에 비례해 처리 속도 향상 및 시스템 안정성 확보

3. 증분 색인 파이프라인

데이터 아키텍처 및 시퀀스

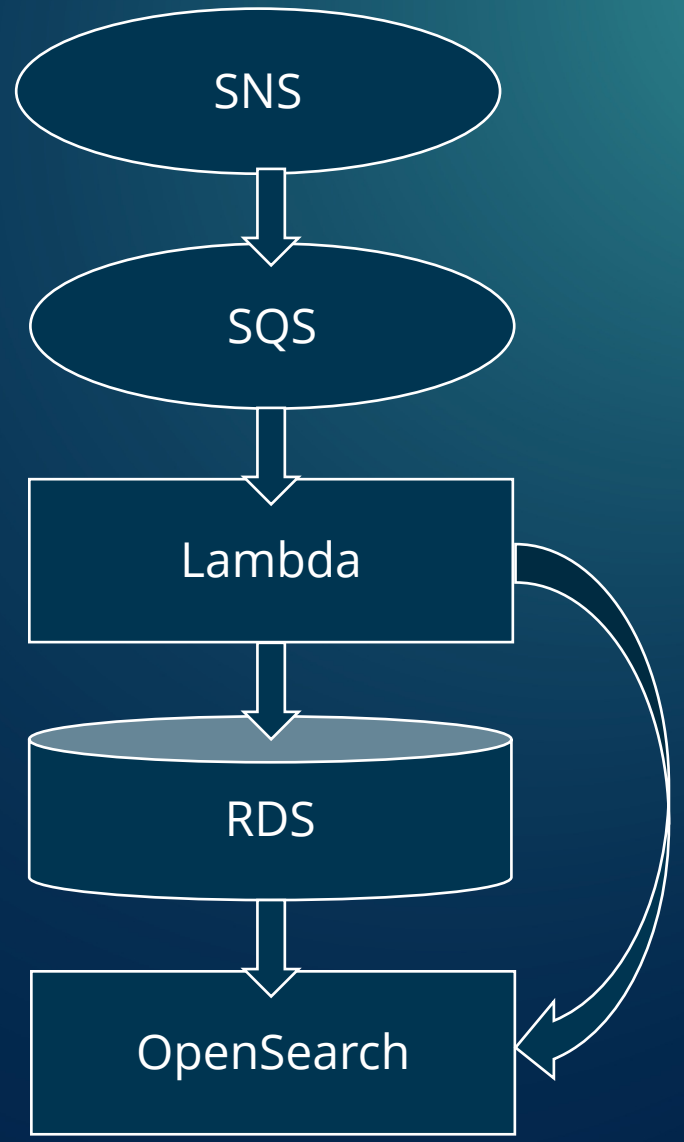
OpenSearch

활용되는 **AWS** 서비스

Service	Description
Amazon Simple Notification Service (SNS)	이벤트 중심의 서버리스 애플리케이션 간에 처리량이 많은 푸시 기반의 다대다 메시징을 제공
Amazon Simple Queue Service (SQS)	소프트웨어 구성 요소 간에 어떤 볼륨의 메시지든 전송, 저장 및 수신 가능
AWS Lambda	이벤트에 대한 응답으로 코드를 실행하고 컴퓨팅 리소스를 자동으로 관리하는 컴퓨팅 서비스로, 아이디어를 최신 프로덕션 서버리스 애플리케이션으로 전환

증분 색인 시퀀스

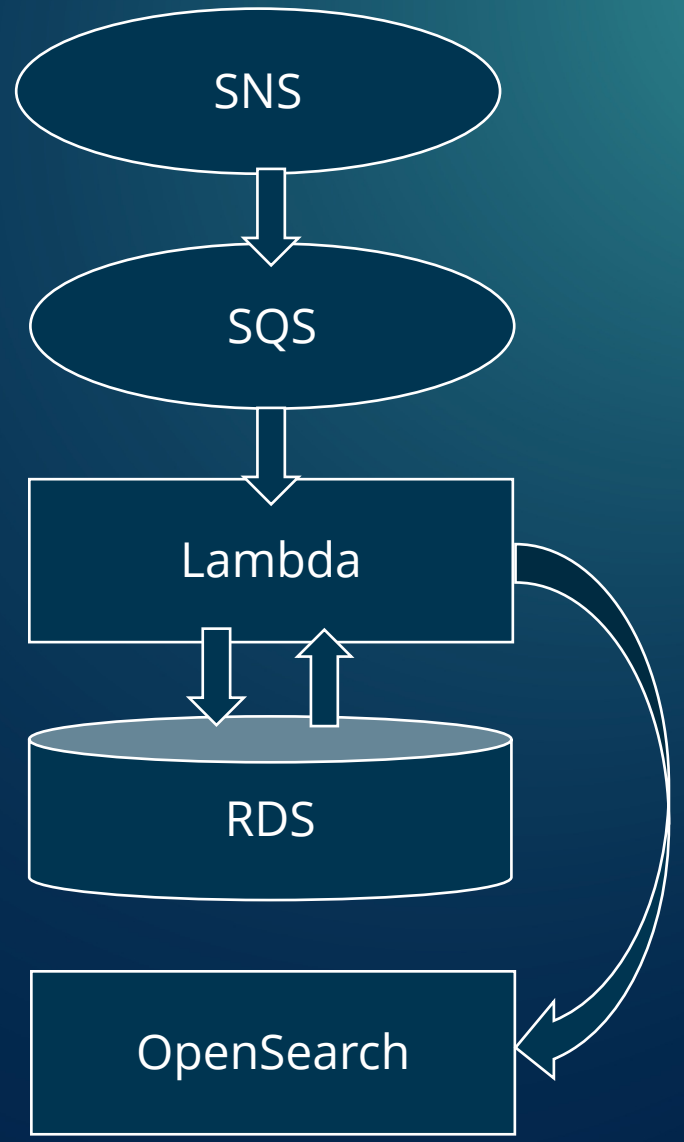
1. 데이터 변경 이벤트 발생
2. SNS 이벤트 게시
3. SQS 큐 추가
4. Lambda 실행



OpenSearch

데이터 추출 및 변경

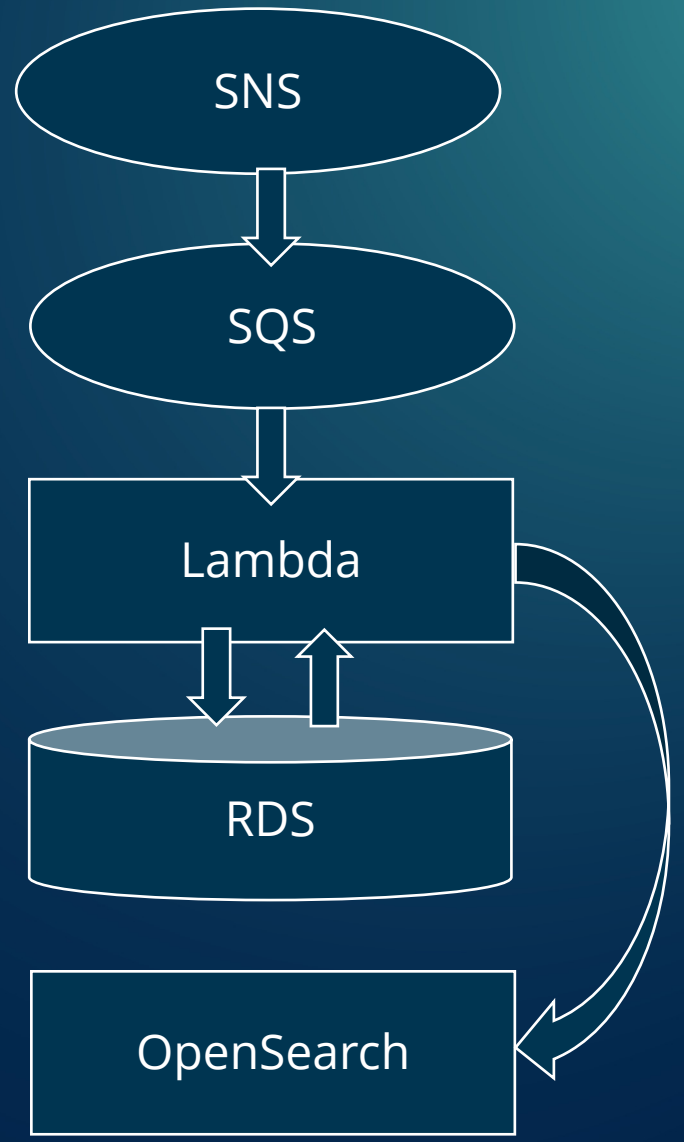
1. SNS, SQS
   - 변경된 데이터가 SQS에 메시지 큐 이벤트 형태로 추가
2. Lambda
   - 큐에 추가된 데이터를 감지 후 실행
   - RDS에 쿼리를 실행해 추가 정보 조회
   - 비즈니스 로직 추가 후 데이터프레임 생성



OpenSearch

데이터 색인

1. Lambda
   - 생성된 프레임을 OpenSearch에 전달해 색인 수행



OpenSearch

추가 고려 사항

- **대량의 변경 데이터 처리 부담**
  - 변경 데이터가 천 단위 이상 발생할 경우 데이터베이스 과부하 가능성 증가
  - CDC(Change Data Capture) 방식을 활용해 색인 구조 개선 필요
- **Lambda 함수의 동시 실행 제어**
  - 메시지큐 이벤트를 **청크 단위(chunking)** 로 분할하여 쿼리 전송
  - Lambda 동시 실행 옵션을 조정해 부하 최소화

주요 이점

- **증분 색인 주기 단축**
  - 기존 30분에서 **1분 이내** 로 단축
  - 실시간 검색 환경에 가까운 성능 구현
- **서버리스 구조 도입으로 운영 효율성 강화**
  - Lambda 기반 서버리스 구조로 인프라 관리 부담 감소
  - 빠른 데이터 처리 및 운영 최적화

검색 API

검색 질의 및 결과 반환 과정

OpenSearch

검색 **API** 도입 배경

- **복잡한 쿼리 추상화**
  - 검색 API가 복잡한 쿼리를 대신 처리하여 사용자의 편의성 향상
  - 개발자가 직접 검색엔진 구조를 이해할 필요 없이 API 호출만으로 데이터 조회 가능
- **시스템 확장성 확보**
  - 검색 API를 통해 검색 서비스의 유연성과 확장성 강화
  - 새로운 기능 추가 및 검색 로직 변경 시 API 단에서만 수정 가능
- **검색엔진 클러스터 변경 시 유연한 대응**
  - 클러스터가 변경되더라도 API 클라이언트만 수정하면 됨
  - 기존 사용자는 코드 변경 없이 검색 기능 유지 가능

검색쿼리 vs 추상화된 검색 API

```
{
  "from": 0,
  "size": 10,
  "sort": [
    {
      "time": "desc"
    },
    "_score"
  ],
  "query": {
    "bool": {
      "must": [
        {
          "multi_match": {
            "query": "□□□",
            "fields": [
              "index.text",
              "index.keyword"
            ],
            "operator": "and"
          }
        }
      ],
      "filter": {
        "bool": {
          "must": [
            {
              "bool": {
                "should": [
                  {
                    "match": {
                      "category_id": "1"
                    }
                  },
                  {
                    "match": {
                      "category_id": "2"
                    }
                  }
                ],
                "minimum_should_match": 1
              }
            },
            {
              "range": {
                "price": {
                  "gte": 1000,
                  "lt": 20000
                }
              }
            },
            {
              "match": {
                "id": "11111"
              }
            }
          ]
        }
      }
    }
  }
}
```

검색쿼리
깊이가 깊고 복잡

```
{
    "from" : 0,
    "size" : 10,
    "query": "□□□",
    "filter" : {
        "category": {
            "id" : ["1", "2"]
        },
        "id" : "11111",
        "price" : {
            "min" : 1000,
            "max" : 20000
        }
    },
    "sort" : "time"
}
```

검색 API
파라미터가 추상화되어 사용자 편리성 증대

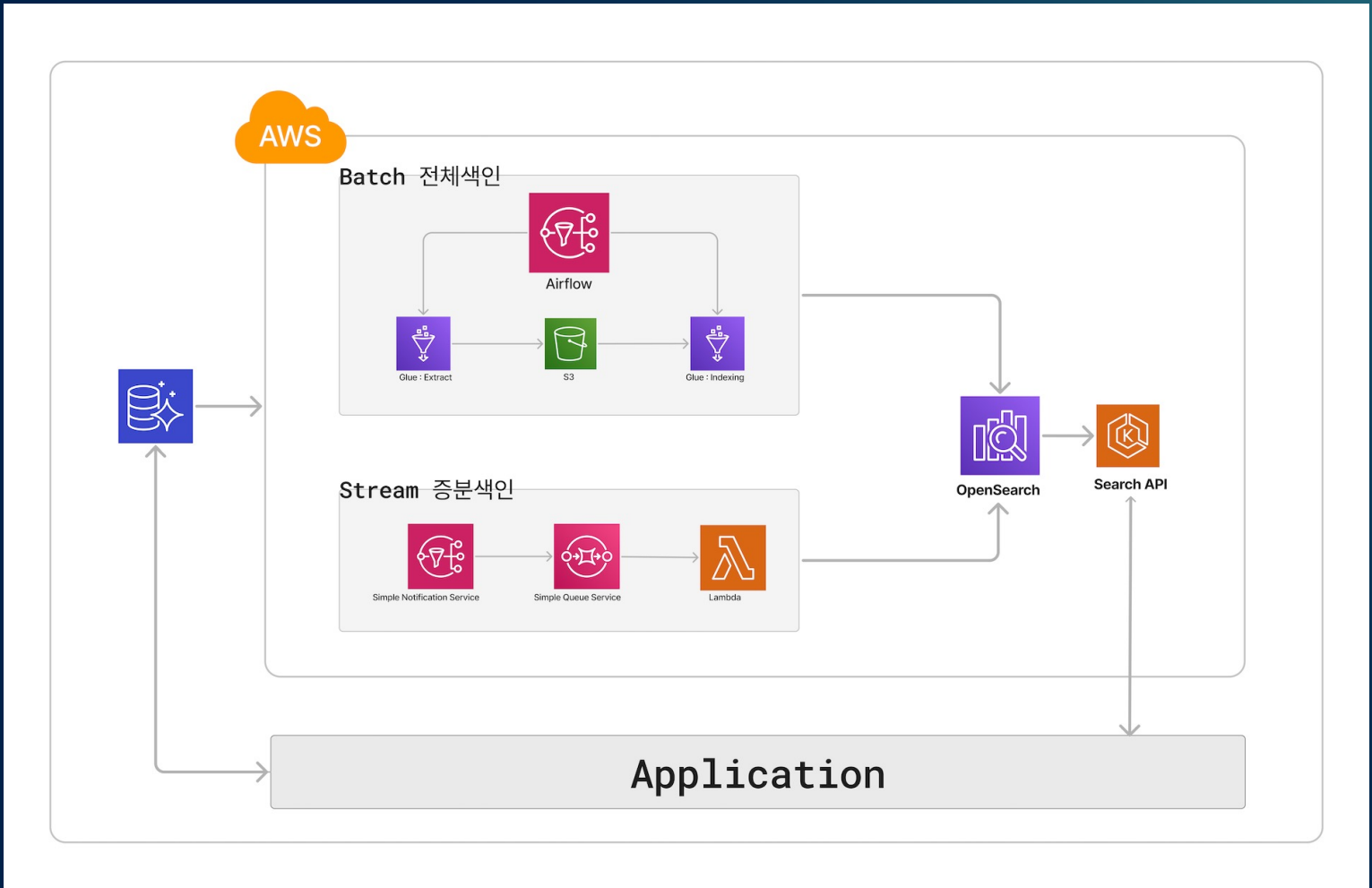
OpenSearch

전체 구성도

전체 색인과 증분색인 통합 구성도

OpenSearch

전체 구성도



OpenSearch

사전운영 및 분석흐름

검색엔진의 분석 흐름에 따른 사전 운영

OpenSearch

토크나이징과 분석기

- 텍스트 -> 토크나이징 -> 분석기 -> 인덱싱
  - 검색의 메타데이터는 토크나이징의 절차를 거쳐 '쪼개지고' 분석기에서 '분석'하여 인덱싱 됨
- 토크나이징에서 문장을 의미 있는 단위로 쪼갬
  - 어절 단위 예시) '검색엔진은 텍스트를 분석한다' -> [검색엔진은, 텍스트를, 분석한다]
- 분석기에서 쪼개진 단어를 분석하여 어간, 어미 등을 삭제함
  - Nori Analyzer 예시) [검색엔진, 텍스트, 분석, 하]

사용자 사전

- 단어가 '쪼개지는 단위'를 조정하는 역할
- 신조어, 비즈니스 용어 등을 도메인에 적합하게 추가하는 용도
- 예시)

  메타데이터 -> 필수템
  토크나이징 결과 ->필수, 템
  사용자 사전 -> '필수템'

동의어 사전

- 동일하거나 유사한 의미의 단어를 서로 매핑하여 다양한 표현으로 입력해도 원하는 결과가 나오도록 함
- 분석 단계 (토큰필터)에서 적용됨
- 예시) 후드티, 후드티셔츠, 후디
- 동의어에 입력된 모든 단어는 사용자 사전에 등록 되어 있어야 오류 확률이 줄어듬

Q & A

OpenSearch