DE31 Team3

문서데이터 수집 및 저장 파이프라인 구축

2024-09-25
팀장: 유정연
팀원: 김대건
박성우
이서연
최성현

Index

1. 프로젝트 개요

IT 시장에서 구직자에게 요구하는 개발 기술은 다양하며, 다른 시장들에 비해 짧은 주기로 기술 트렌드가 변화합니다. 이러한 시장 배경은 구직자는 시시각각 변화하는 요구 기술 스택을 파악하기 어려우며, 직무에서 요구되는 직무 기술 스택을 파악하는데 많은 시간을 허비합니다. 저희는 IT 구직자들에게 기술 스택 파악에 필요한 정보를 제공하여 이러한 구직 문제를 완화하고자 TMI(Tech Map IT, TMI) 프로젝트를 진행하였습니다.

TMI 프로젝트는 주요 채용 사이트에서 수집한 데이터를 바탕으로, 구직자들에게 공고에 명시된 요구 기술 목록을 제공하며, 나아가 기술 스택의 빈도수를 분석하여 공고 중인 직무에서 자주 요구되는 기술들에 대한 지표를 사용자들에게 제공합니다.

2. 프로젝트 요구사항

본 프로젝트의 주 목적은 데이터 제공을 위한 안정적인 데이터 수집과 지속적인 배포이며, 이 목적을 달성하기 위해 요구되는 사항으로는 서비스 기능으로 요구되는 `서비스 요구사항`과 원활한 서비스 운영을 위해 요구되는 `기술 요구사항`이 존재합니다:

서비스 요구사항

- **공고 정보 제공**: 수집한 공고 데이터를 웹을 통해 사용자에게 제공.
- **직무 기술스택 정보 제공**: 공고 내 직무에서 요구되는 기술 목록을 추출하여 사용자에게 제공.
- **기술 스택 시각화 제공**: 추출한 기술 스택을 기반으로 시각화 기능 제공.

기술 요구사항

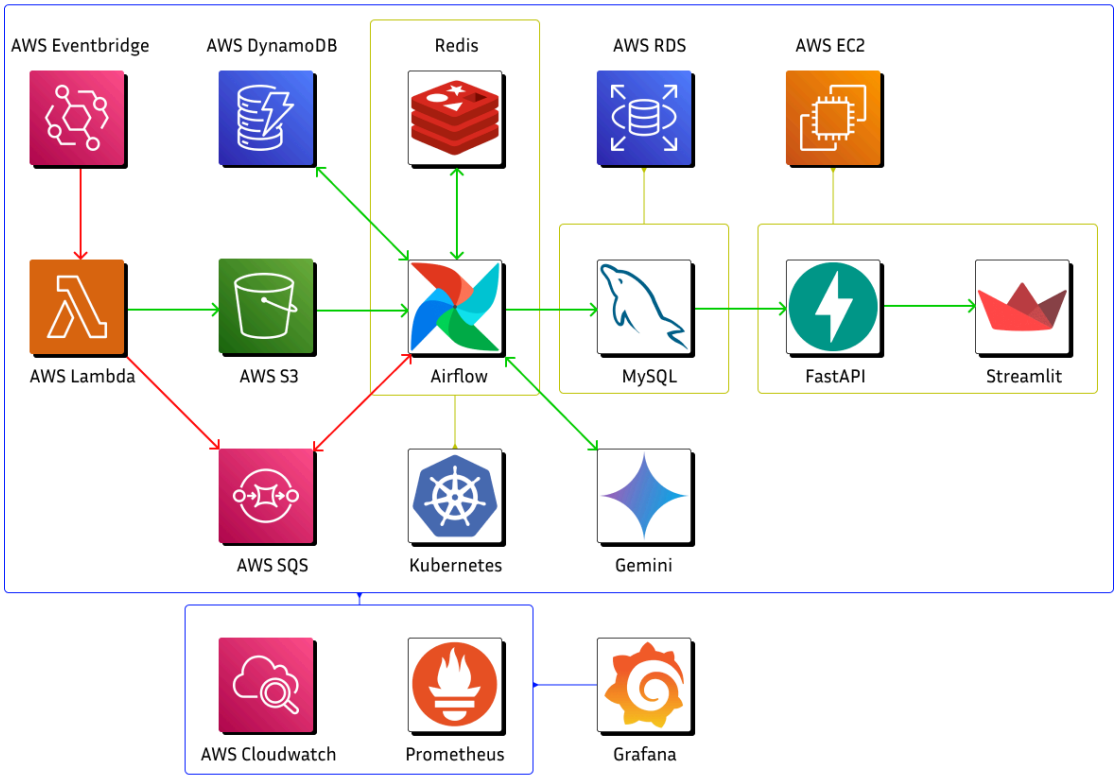
- **데이터 수집 기능 항상성 보장**: 서비스 시스템의 장애가 발생하더라도 데이터 수집 기능 보장.
- **메달리온 아키텍처 기반 데이터 파이프라인**: 데이터 질적 수준을 기반으로 데이터 파이프라인 구축.
- **시스템 확장성 고려**: 채용 공고 데이터 수집 기능 확대 및 시스템 확장을 고려하여 설계 및 CI/CD 기능 제공.
- **시스템 모니터링 및 장애 대응**: 유지보수 및 관리를 위해 로그 기반 시스템 모니터링 기능 제공 및 시스템 장애 발생시 빠른 복구 가능.
- **비용 관리**: Cloud 서비스 사용시 발생하는 비용을 최소화.

3. 개발 환경 및 사용된 기술 스택

범위	기술명	비고
Cloud / Serverless	AWS EventBridge	주기성 Lambda 데이터 수집 이벤트 활성화
	AWS Lambda	서버리스 환경에서 공고 데이터 수집
	AWS S3	수집한 원시 데이터 저장
	AWS SQS	데이터 파이프라인 간 이벤트 연동
	AWS CloudWatch	AWS 관련 로그 수집 메트릭
	AWS DynamoDB	전처리 및 파생한 공고 데이터 저장
	AWS RDS(MySQL)	배포용 공고 데이터를 관계형 데이터로 저장
	AWS EC2	웹 서비스 배포용 클라우드 서버 구축
	Streamlit	웹 사이트 구축
	FastAPI	RESTful API를 통한 데이터 배포
	Gemini LLM API	파생 데이터 생성 및 추출
On-Premise	Kubernetes	컨테이너 기반 분산 시스템 구축
	Redis	중복처리를 위한 캐싱 DB
	Airflow	데이터 파이프라인 오케스트레이션 및 Dag CI/CD
	Prometheus	AWS 외 로그 수집 메트릭
	Grafana	시스템 현황 모니터링 시각화
External Packages	Pandas	데이터 정제
SCM & CI / CD	Github Actions	웹 CI/CD 기능 제공
	Github Project	프로젝트 협업용 일정 및 업무 관리

4. 프로젝트 아키텍처 및 파이프라인 흐름

다음 이미지는 전체적인 프로젝트 아키텍처의 간략화된 이미지입니다.



본 프로젝트는 Kubernetes 기반 컨테이너 분산 시스템과 AWS 기반 서버리스 환경에서 구축되었으며, Airflow를 통해 전체 데이터 파이프라인을 관리합니다. 아키텍처의 흐름은 다음과 같습니다:

1. **데이터 수집 단계**: AWS Lambda와 AWS Eventbridge로 채용 사이트에서 공고 데이터를 주기적으로 수집하고 S3에 저장합니다. 이 과정에서, 수집의 결과 및 완료 여부를 SQS를 통해 Airflow에 전달해 DAG를 작동시킵니다.
2. **데이터 정제 단계**: Lambda를 통해 수집되어 AWS S3에 저장된 데이터는 Medallion 아키텍처를 토대로 데이터의 질적 수준에 따라 단계별로 정제됩니다. Pandas를 통해 1차 전처리 작업을 진행한 후, Gemini LLM을 통해 파편화되어 있는 비정형적인 정보를 정형화된

데이터로 2차 전처리 작업을 진행해 DynamoDB에 저장합니다. 최종적으로 관계형 데이터베이스에 적합하게 3차 전처리 작업을 진행한 뒤 RDS에 저장하는 것으로 정제가 완료됩니다.

3. **시각화 단계**: AWS RDS로 구축된 MySQL 데이터베이스로부터 FastAPI와 sqlalchemy를 통해 RESTful API를 구현해 백엔드 서버를 구축했습니다. 이 API 서버와 통신하며 최종적으로 데이터를 받아 사용자에게 시각화하는 프론트엔드 서버는 Streamlit를 통해 구축했습니다. 프론트엔드 서버에서는 사용자에게 채용 공고 데이터를 제공하고, 빈도가 높은 상위 10개의 기술 스택에 대한 시각화 지표를 제공합니다.
4. **모니터링 및 유지보수 단계**: AWS CloudWatch를 통해 Cloud 상으로 구축된 파이프라인 구성 요소에 대한 로그를 수집하고, Prometheus를 통해 On-Premise Kubernetes로 구축된 파이프라인 구성 요소들에 대한 로그를 수집하여 Grafana를 통해 전체 파이프라인을 모니터링하기 위한 대시보드 웹 인터페이스를 제공합니다. 또한, Airflow DAG를 통해 대부분의 파이프라인 핵심 이벤트들이 호출되기 때문에 Airflow의 웹 서버를 통해 각 DAG의 동작을 모니터링합니다.

   파이프라인의 유지보수에 핵심적인 요소는 총 4가지로, AWS Lambda를 통해 구현된 크롤러 코드, 전체 오케스트레이션을 위한 Airflow DAG 코드, 웹 인터페이스의 백엔드 단계에 해당하는 FastAPI 코드, 웹 인터페이스의 프론트엔드 단계에 해당하는 Streamlit 코드입니다. 이 코드들을 지속적으로 유지/보수하기 위해 Airflow의 Git Sync 기능과 Github Actions를 활용해 CI / CD 파이프라인을 구축했습니다.

5. 향후 계획

본 프로젝트의 서비스를 발전시키기 위한 고도화 작업 및 구현해야 될 기능은 다음과 같습니다:

- **데이터 수집 소스 확대:** 기존 네 개의 채용 사이트 외 다른 채용 사이트에 대한 데이터 수집.
- **데이터 생명 주기 기능**: 마감된 공고 데이터를 관리 및 데이터 거버넌스 정책 설정.
- **데이터 분석 기능**: 기술 스택 빈도수 분석 외 여러 분석 및 시각화 지표 제공.
- **세부 검색 기능**: 공고 데이터 검색에 대한 여러 검색 필터 제공 및 검색 기능 성능 개선.
- **추천 성능 개선**: 단순한 빈도수 Top 10을 제공하는 대신 추천 알고리즘을 도입하여 사용자가 제공한 검색 키워드에 적절한 기술 스택의 목록 제공.
- **웹 인터페이스 개선**: 프론트엔드 서버를 Streamlit이 아닌 Django로 리팩토링 하여 웹 인터페이스 개선.

위 기능을 추가하면 사용자는 데이터 수집의 확장과 생명 주기 관리를 통해 더 나은 데이터 품질과 신속한 분석을 경험할 수 있습니다. 추가적으로 세부 검색 기능과 검색 성능 개선은 사용자가 보다 정밀하고 효율적인 데이터 검색을 할 수 있게 도와줄 것입니다. 마지막으로 웹 인터페이스 개선은 사용자 경험을 향상시켜 사용 편의성을 극대화할 수 있을 것으로 예상합니다.