LLM(Large Language Model)을 사용한 AI 챗봇 연구

팀 명: **ForPaw**

부산대학교 **정보컴퓨터공학부**

202155595 이한홍
201924475 박재홍
202155590 이종일

지도교수: 김호원

목차

1. 과제 목표

본 졸업 과제는 LLM 을 활용하여 로그데이터를 주기적으로 분석하는 시스템을 개발하는 것이다.

- 클라우드에서 실행되고 있는 Docker 컨테이너들의 로그를 Logstash, FileBeat, ElasticSearch 를 사용해서 로그를 주기적으로 수집한다. 예를 들어, 웹 서버스를 운영중이라면 Spring Boot, React, Nginx, MySQL 등의 어플리케이션이 실행되고 있는 컨테이너를 로깅한다.
- Metricbeat, ElasticSearch 를 사용해서 클라우드에서 동작하고 있는 리눅스 인스턴스(EC2 인스턴스)의 성능 지표를 주기적으로 수집한다.
- ElasicSearch 에 저장된 로그들을 LLM 을 활용하여 분석/요약하고 비정상적인 활동이나 오류를 탐지한다.
- 비정상적인 활동이나 오류가 탐지 됐을 때, Slack API 나 Google Mail API 를 사용하여 관리자에게 알람이 전송되도록 개발한다.
- 분석/요약된 결과 혹 수집된 로그들을 웹을 통해서 조회 할 수 있도록 웹 페이지를 개발한다.

2. 제약사항 및 수정사항

2.1 로그 관련 제약사항

(1) 로그 수집

- 런타임에 실행되고 있는 각 Docker 컨테이너로부터 로그를 중앙화 된 장소로 전송해야 하는데. 어떤 기술을 사용하여 데이터베이스까지 전달할지 선택에 어려움이 있다.
- Docker 컨테이너에서 생성된 로그를 어떤 주기로, 어떤 전처리를 거쳐서, 어떤 형식으로, 어떤 데이터베이스에 저장할지 결정하는 데 어려움이 있다.

(2) 로그 분석

- 어플리케이션마다 로그의 형식이 모두 다르기 때문에, 이들을 일괄적인 방법으로 분석하는 것에 어려움이 있다.
- LLM 이 분석할 수 있는 토큰 수는 한정되어 있기 때문에, 토큰 수를 넘는 로그가 생성됐을 때 처리에 어려움이 있다.

2.2 LLM 모델 제약 사항

(1) Context-length

- 기존에 사용하기로 한 Llama3 - 8B 의 경우 성능도 준수하고 비용도 저렴하지만, context-length 가 8k 이기 때문에 로그의 양이 많아지면 처리할 수 없는 제약사항이 있다. 그래서 비슷한 경령화 모델이지만 context-length 가 128k 고, 성능도 더 뛰어난 Chat GPT 4o-mini 로 대체되었다.

(2) 파인 튜닝

- 로그를 바탕으로 비효율적인 패턴을 탐지하기 위해서 파인 튜닝을 할 계획이었지만. Chat GPT 4o-mini 의 경우 이미 이런 패턴들을 꽤 학습한 모델이기 때문에, 테스트 시 효율적으로 패턴을 감지하는 결과를 보여서 파인튜닝보다 프롬프트 엔지니어링에 초점을 맞추기로 계획을 변경하였다.

2.3 데이터 베이스 제약사항

(1) RDBMS

- 기존에는 Logstash 를 통해 수집한 데이터를 가공해서 RDBMS 인 MySQL 에 저장해서 활용할 계획이었다. 하지만 방대한 양의 로그 데이터를 저장하고 조회해야 하는 과제의 성격과 맞지 않고 Logstash 와의 연동성도 좋지 않기 때문에, 로그 저장소는 분산형 검색 엔진 DB 인 ElasticSearch 로 대체되었다.

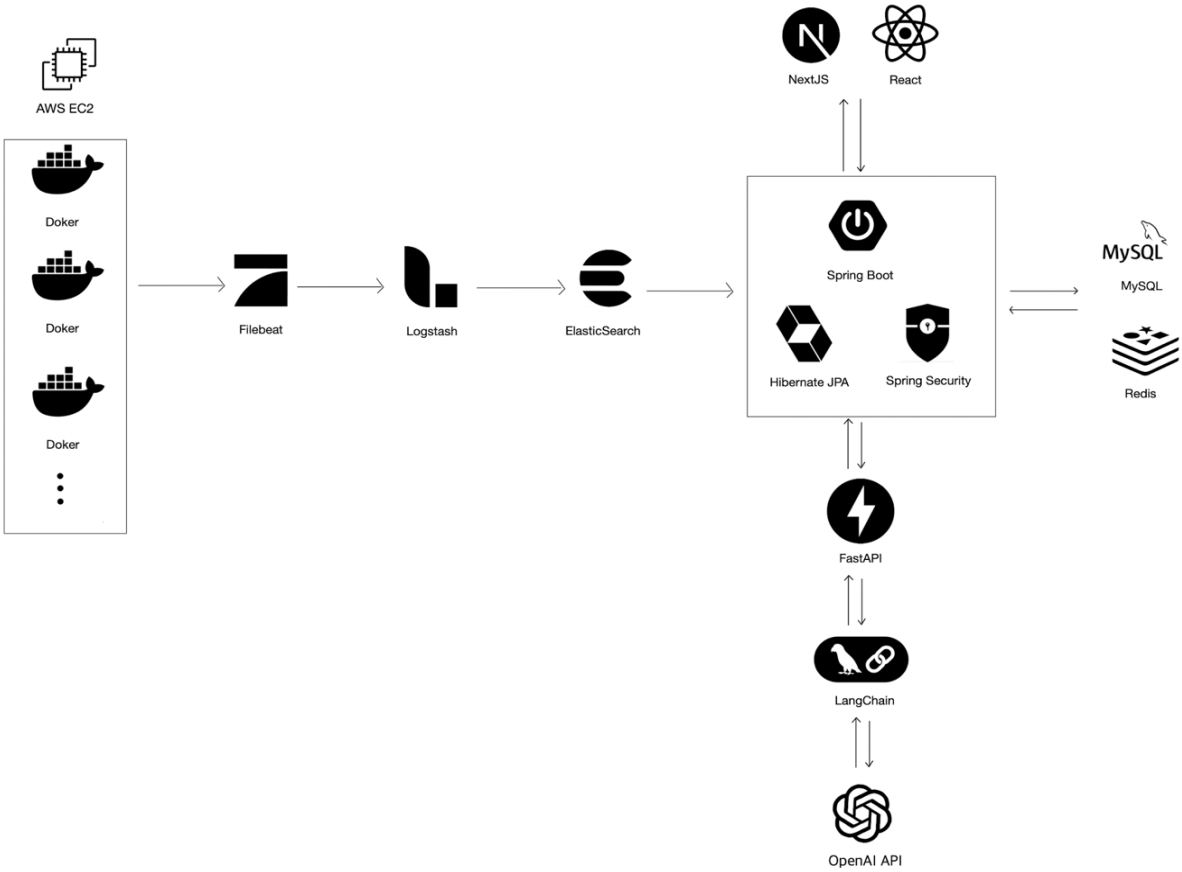
2.4 시스템 리소스 제약사항

(1) AWS EC2

- 기존에는 하나의 EC2 인스턴스에서 어플리케이션 컨테이너와 로깅 기술들을 함께 작동시킬 계획이였으나. 기존의 어플리케이션의 성능에 영향을 줄 수 있기 때문에, 로깅 기술들은 별도의 EC2 에서 작동시키는 것으로 수정하였다.

3. 설계 상세화

3.1 전체 구성도



요소	내용
로그 스택	Filebeat 를 사용하여 로그를 로그를 수집하면, Logstash 가 수집된 로그를 전처리해서 ElasticSearch 에 저장한다.
백엔드 서버 (Spring)	ElasticSearch 에 저장된 로그들을 API 형태로 클라이언트에 전송하는 역할을 하고. 또 LLM 을 활용한 분석을 위해 주기적으로 FastAPI 에 로그들을 전송한다.
백엔드 서버 (FastAPI)	Spring 으로부터 전달받은 로그 데이터들을 LangChain 과 OpenAI API 를 사용하여 분석한다.
프롬프팅	OpenAI API 를 호출하기 전, LangChain 을 사용하여 프롬프팅 값을 추가 한다.
DB (MySQL)	웹 페이지 인증에 사용할 사용자 정보나 화면에 필요한 데이터들을 저장한다.
DB (ElasticSearch)	로그 데이터들을 저장한다.
프론트엔드	React 나 NextJS 를 사용하여 사용자에게 보여줄 대시보드나 로그 데이터들을 웹 형식으로 보여준다.

3.2 Front-End

- React 와 NextJS 를 사용하여 Front-End 를 구현한다.
- 주요 페이지로는 대시보드, 각 컨테이너별 로그 분석/요약 페이지 ,각 컨테이너별 로그 상세 페이지, 알림 내역 페이지가 있다.

3.1 Back-End

- Spring Boot 3.2 와 FastAPI 0.1 을 사용한다.
- Spring Boot 와 FastAPI 간 통신이나 Front-End 통신 시 REST API 방식으로 통신한다.
- 빠른 배포/테스트를 위해 Docker 컨테이너 형태로 실행한다.
- Hibernate JPA 를 통해 ElasticSearch 나 MySQL 에 접근한다.
- Redis 를 사용해 스트리밍과 세션을 구현한다.

3.1 데이터베이스 설계

Spring Boot 3.

4. 수정된 과제 추진 내역

<table>
<tr><th colspan="5">6 월</th><th colspan="4">7 월</th><th colspan="5">8 월</th><th colspan="4">9 월</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>1</th><th>2</th><th>3</th><th>4</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td colspan="5" rowspan="2">요구 사항 분석 후
초기 기획
및 아키텍쳐 설계</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="3" rowspan="2">Docker 학습</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="4">ELK 스택 학습</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="4">LangChain 학습</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="4">FastAPI 학습</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5">ELK 파이프라인 구축</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5">LangChain 구축</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="7">스프링 부트 API 개발</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="7">FastAPI API 개발</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">UI 개발</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">프롬프팅 진행</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>배포</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td rowspan="2">발표 준비</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

5. 구성원별 진척상황

이한홍	**설계** - API 설계 - DB 스키마 설계 **로깅 시스템 구축** - Filebeat => Logstash => ElasticSearch 로 이어지는 파이프라인 구축 - 로그 데이터 전처리/후처리 로직 구현 - 로그 조회를 위한 ElasticSearch 와 Spring Boot 연동 - 로깅을 위한 EC2 인스턴스와 어플리케이션 EC2 인스턴스 연동 **BackEnd (Spring Boot)** - 유저 인증을 위한 Spring Security 구현 - ElasticSearch 에 저장된 로그를 조회하기 위한 JPA 쿼리 메서드 구현 - API 를 통한 Spring Boot 와 FastAPI 의 통신 구현 **BackEnd (FastAPI)** - Langchain 연동 - OpenAI API 연동 **인프라** - AWS EC2 생성 및 Docker 설치 - Jenkis 를 활용한 CI/CD 파이프라인 구축
박재홍	**프롬프트** - 로그 데이터의 프롬프트 연구
이종일	**FrontEnd** - 화면 설계

S3 를 활용하여 로그 데이터를 통합 저장하며, 로그 데이터의 형식과 구조를 표준화하여 일관된 분석이 가능하도록 전처리한다.

(2) LLM 모델 학습 및 최적화

다양한 비정상적인 활동 패턴(비정상적인 로그인 시도, 데이터 유출 시도 등)을 포함한 로그 데이터를 사용하여 LLM 을 학습시키고, 학습된 모델을 클라우드 환경과 운영 환경에 맞게 최적화하여 높은 정확도로 비정상적인 패턴을 탐지할 수 있도록 개선한다.

(3) 실시간 분석 시스템 구현

실시간으로 로그 데이터를 분석하여 비정상적인 활동을 탐지하는 시스템을 구축하고, LLM 의 예측 결과를 기반으로 실시간 경고 및 알림을 생성하여