

SK네트웍스 Family AI 과정 12기

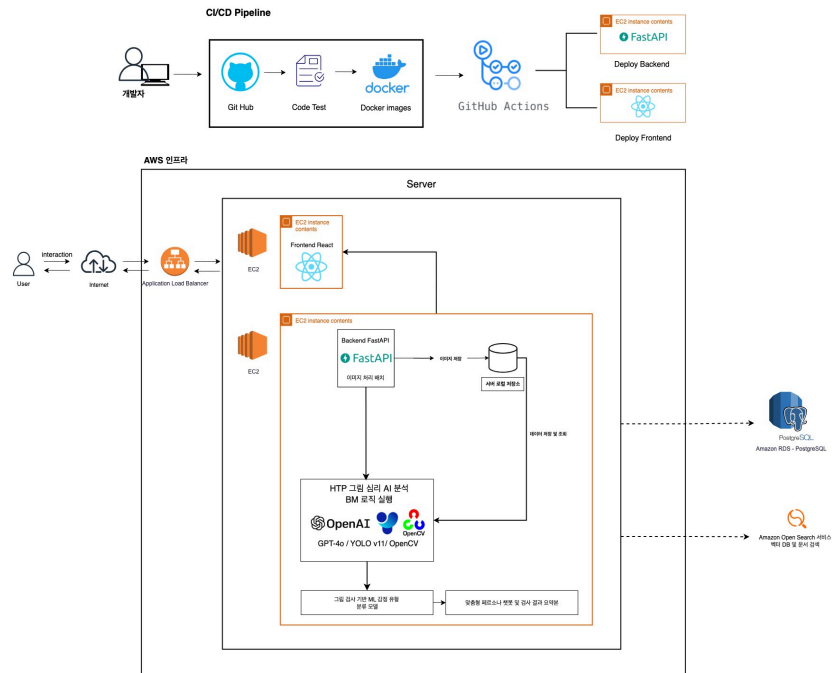
모델배포 시스템 구성도

산출물 단계	모델배포
평가 산출물	시스템 구성도
제출 일자	2025.07.25
깃허브 경로	https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-6TEAM
작성 팀원	김승학, 이지복

*아키텍처, 구성도 등 도식화 흐름 이미지 삽입 및 부가설명

개요

● AWS 시스템 구상도



구성 요소

- **Backend with FastAPI**

1. Backend CI/CD Pipeline :

- GitHub Repository : 소스 코드 버전 및 관리 및 저장소
- GitHub Actions : 자동화된 빌드, 테스트, 배포 파이프라인
- Docker : 애플리케이션 컨테이너 화

2. ECR (Elastic Container Registry) :

Docker 이미지를 저장하는 AWS 서비스.

3. ECS (Elastic Container Service) :

컨테이너화된 애플리케이션을 실행하고 관리

4. ALB (Application Load Balancer) : 트래픽 라우팅 및 HTTPS 연결 지원.

5. RDS with PostgreSQL : Postgresql 데이터베이스를 제공하는 Amazon RDS.

외부 API 호출 및 PDF 파일 저장을 위한 Amazon S3.

- **Frontend with React**

1. React 애플리케이션

- 사용자 인터페이스

2. EC2 인스턴스

- Frontend 호스팅 서버

Backend Infrastructure

1. **FastAPI**: Python 기반 REST API 서버
2. **EC2 인스턴스**: Backend 애플리케이션 서버
3. **PostgreSQL (RDS)**: 관계형 데이터베이스
4. **Amazon OpenSearch**: 검색 및 분석 엔진

AI/ML Services

1. **OpenAI GPT-4:** 자연어 처리 및 생성
2. **YOLO v11:** 객체 탐지 모델
3. **OpenCV:** 컴퓨터 비전 라이브러리

External Services

1. **HTP 그룹 심리 AI 분석:** 외부 심리 분석 서비스
2. **맞춤형 챗봇:** 개개인별 맞춤형 페르소나 챗봇 제공

데이터 흐름

개발 및 배포 흐름

1. 개발자가 GitHub에 코드 커밋
2. GitHub Actions가 자동으로 트리거
3. 코드 테스트 및 Docker 이미지 빌드
4. ECR에 Docker 이미지 푸시
5. EC2 인스턴스에 새 이미지 배포

사용자 요청 처리 흐름

1. 사용자가 웹 브라우저에서 애플리케이션 접근
2. Internet을 통해 ALB로 요청 전달
3. ALB가 Frontend EC2 인스턴스로 라우팅
4. React 애플리케이션이 사용자 인터페이스 제공

Backend API 처리 흐름

1. Frontend에서 Backend FastAPI로 API 요청
2. FastAPI가 요청을 처리하고 필요시 다음 서비스 호출:
 - PostgreSQL에서 데이터 조회/저장
 - OpenAI API 호출 (GPT-4)
 - YOLO v11으로 이미지 분석
 - OpenCV로 이미지 처리
3. 외부 심리 분석 및 맞춤형 챗봇 시스템과 연동
4. 처리 결과를 Frontend로 반환

데이터 저장 및 검색 흐름

1. 애플리케이션 데이터는 PostgreSQL에 저장
2. 검색 기능은 Amazon OpenSearch 활용
3. AI 분석 결과는 데이터베이스에 저장
4. 심리 분석 및 선택된 챗봇 데이터는 별도 처리