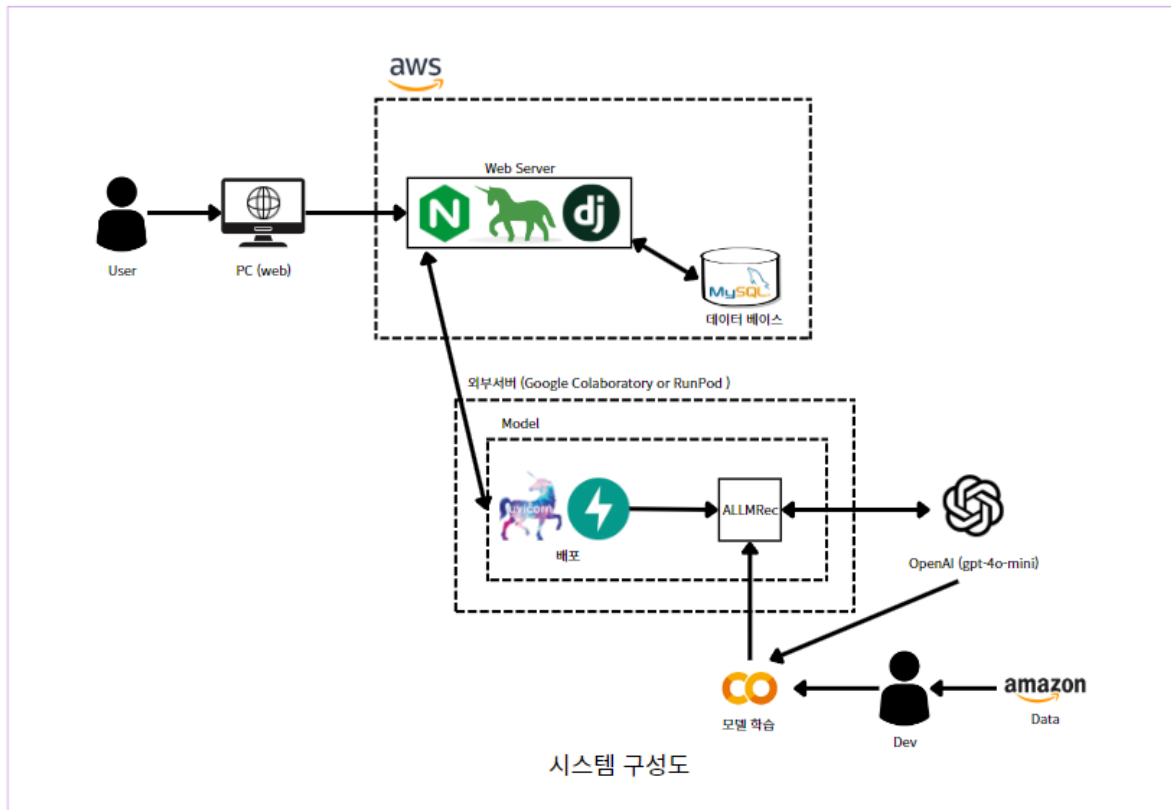


시스템 아키텍처



1. 개요

이 시스템은 AWS 클라우드 인프라를 기반으로 Django 웹 애플리케이션 서버가 EC2 인스턴스에서 구동되며, 데이터베이스는 AWS RDS를 통해 관리, 외부 서버인 Google Colab 혹은 Runpod에서 실행되는 AI 모델과 api로 연동하여 데이터 관리와 AI 응답을 생성 상품을 추천함

2. 구성요소

1. 엔드 유저

- **User (사용자):** 시스템의 최종 사용자, 웹 브라우저를 통해 시스템에 접근하여 질문 입력.
- **PC (web):** 사용자가 시스템에 접속하는 단말기. 웹 브라우저를 통해 시스템에 접근하여 Django 웹 애플리케이션을 통해 상호작용.

2. Web Server

- **Nginx:**
 - 웹 서버 역할로 사용자 요청을 처리하여 Gunicorn으로 전달.
 - 정적 파일을 제공하고 트래픽을 관리하는 역할을 수행하여 서버 부하를 분산.
- **Gunicorn:**
 - Django 애플리케이션을 실행하는 WSGI 서버로, Nginx로부터 받은 요청을 Django에 전달하고 응답을 생성.
 - 동시 요청 처리 성능을 최적화하여 Django 애플리케이션의 안정적인 구동을 지원
- **Django:**
 - 웹 애플리케이션의 서버 사이드 로직을 처리하고 사용자 요청을 라우팅하며, 사용자의 입력 처리하고 AI 모델로 전송하며 결과 반환

3. 데이터베이스 (AWS RDS)

- **MySQL:** 데이터베이스로, 사용자 정보, 기록 데이터, 아이템/리뷰 등의 원본 및 처리된 데이터를 저장 Django 애플리케이션에서 필요한 데이터 조회와 저장을 지원

4. Model

- 외부 연산 서버인 Google Colab 혹은 RunPod에서 실행
- **ALLMRec:** 추천 시스템의 코어 모델로, 사용자 데이터와 IIm을 활용하여 맞춤형 추천 제공
- **Uvicorn**과 **FastAPI:** 외부 연산 서버에서 **Uvicorn** ASGI 서버와 **FastAPI**를 이용해 ALLMRec 모델의 API를 구성
- **ngrok:** 제한된 네트워크 환경에서도 Django 서버와의 연결을 유지하기 위해 **ngrok**을 사용하여 안전한 터널링을 제공. 이를 통해 외부 연산 서버와 Django 서버 간의 안정적인 API 통신 가능.

5. 모델 학습

- **Dev (개발자):** Colab환경에서 Amazon Data 정제 및 모델 활용/학습
- **Amazon Data:** Amazon에서 제공되는 데이터 리뷰 데이터 및 유저 데이터, 모델 학습에 사용

3. 데이터 흐름

1. **사용자 요청:** 사용자가 웹 브라우저를 통해 질문 입력

2. **요청 처리:** Django 애플리케이션에서 외부 연산 모델로 api를 통해 전달
3. **AI 응답 생성:** gpt 40 mini 모델이 연결되어있는 Allmrec 모델이 질문을 처리하여 추천된 상품 답변을 생성
4. **결과 반환:** Django 애플리케이션을 통해 사용자에게 응답 반환