생성형 AI 기반 RAG 개발자 과정

로컬 모델 · 배포 고려

Session 31 / 33 — Day 4

비용 0 · 프라이버시 · 운영 배포

박수현 · (사)한국정보공학기술사회 · 2026

LLM 이론 · LangChain · ChromaDB CRUD · Agent — 4일 핸즈온

출처: Day 4 로컬·배포

이번 시간을 마치면

💻 로컬 모델

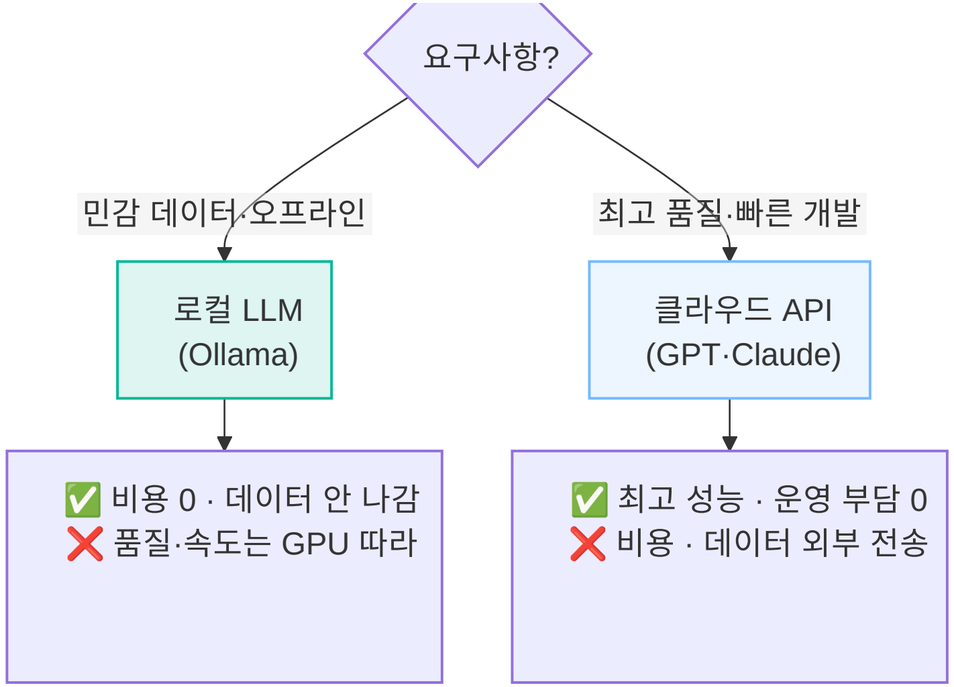
- **Ollama** 로 로컬 LLM RAG
- **양자화**(Q4/Q8) 트레이드오프
- 로컬 임베딩 모델

🚀 배포·운영

- **로컬 vs 클라우드** 선택 기준
- 환경변수·비용·모니터링
- 컨테이너로 띄우기

🎯 목표: 비용·프라이버시 선택지를 파악하고, 서비스를 **로컬 환경 밖**으로 배포할 준비를 갖춘다.

API와 로컬의 선택 기준



💡 단일 정답은 없다 — **민감도·예산·품질** 요구에 따라 선택하며, 하이브리드 구성도 일반적이다.

모델 선언부 한 줄 교체 (스니펫)

```
# 1. 모델 받기 (한 번)
ollama pull qwen2.5:7b
ollama pull nomic-embed-text
```

```
from langchain_ollama import ChatOllama, OllamaEmbeddings

llm = ChatOllama(model="qwen2.5:7b")          # API → 로컬
emb = OllamaEmbeddings(model="nomic-embed-text")
# 나머지 RAG 체인은 그대로 — LangChain 이 추상화
```

- LangChain 추상화로 **모델만 교체**, 체인 코드는 그대로 유지
- 로컬 임베딩(`nomic-embed-text`)으로 검색 비용도 0

> 📁 참고: `7.RAG/7.local_model/1.ollama/`

Q4 / Q8 — 정확도와 메모리의 균형

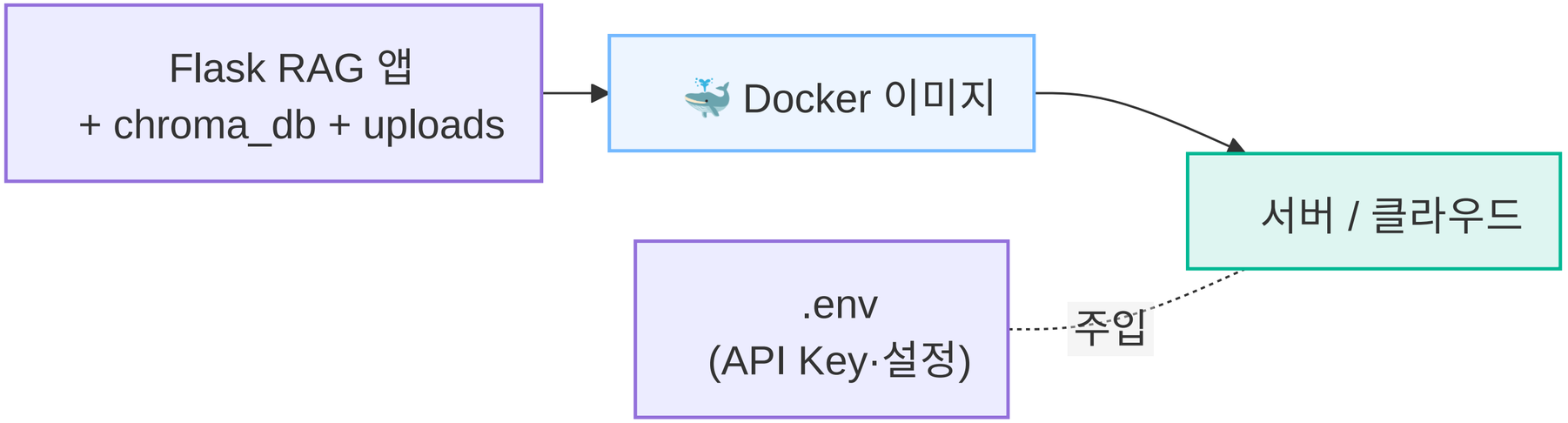
양자화란

- 가중치 정밀도를 낮춰 **크기·메모리 ↓**
- 7B 모델: FP16 ~14GB → **Q4 ~5GB**
- 일반 노트북에서도 구동 가능

수준	크기(7B)	특징
FP16	~14GB	원본 품질
Q8	~8GB	거의 동일
Q4	~5GB	약간 손실, 실용적

> 🔑 노트북 RAG 데모에는 대체로 **Q4 7B**로 충분하다.

컨테이너 기반 이식 배포



점검 항목

- **환경변수**로 키·설정 분리(코드 하드코딩 금지)
- `chroma_db` · `uploads` **볼륨** 영속화
- 의존성 버전 고정(`requirements.txt`)

운영 관점

- **비용** 모니터링 (토큰·호출 수)
- **로그·추적** (LangSmith 등)
- 동시 사용자·속도 측정

데모에서 서비스로 전환할 때

항목	점검
비밀값	`.env`·시크릿 매니저로 분리, 깃 제외
데이터 영속	`chroma_db` · `uploads` 볼륨 마운트
비용	토큰·호출 모니터링 + 상한
안정성	에러 처리·타임아웃·재시도
관측성	요청·도구 호출 로그/추적
모델 선택	작업 난이도별 모델·로컬 혼용

💡 "로컬에서 동작함"과 "서비스로 운영함"의 격차가 이 표에 담겨 있다. 데모 단계에서는 과도하지 않게, 운영 단계에서는 빠짐없이 적용한다.

핵심 5가지

1. ✅ 로컬 LLM(**Ollama**) — 비용 0·데이터 안 나감, 품질은 GPU 의존
2. ✅ LangChain 추상화로 **모델만 교체**하면 로컬 RAG 전환
3. ✅ **양자화**(Q4/Q8)로 노트북에서도 7B 구동
4. ✅ 배포 = **컨테이너 + 환경변수 + 볼륨 영속**
5. ✅ 운영은 **비밀값·비용·관측성**을 빠짐없이