

المحور الثالث: الهيكلية والمكدس التقني

Platform Architecture and Technology Stack

المؤلف: Manus AI
التاريخ: 3 نوفمبر 2025
الهدف: تحديد المكدس التقني (Technology Stack) الأمثل الذي يوازن بين سرعة التطوير، وقابلية التوسيع، والكفاءة المطلوبة لمعالجة الذكاء الاصطناعي الثقيلة . (Heavy AI Processing)

1. الهيكلية العامة (General Architecture)

تعتمد الهيكلية المقترحة على مبدأ **فصل الاهتمامات** (Separation of Concerns)، حيث يتم فصل الواجهة الأمامية (Frontend) عن الواجهة الخلفية (Backend)، ويتم التعامل مع مهام الذكاء الاصطناعي الثقيلة بشكل غير متزامن (Asynchronously) عبر نظام قائمة انتظار.

أ. المكدس التقني الموصى به

المكون	الوصية التقنية	التبrier الاستراتيجي والتقني
واجهة الأمامية (Frontend)	React/Next.js	يوفّر مجتمعاً ضخماً، أدوات قوية، وقابلية ممتازة للتوسيع. Next.js يضيّف ميزات مثل العرض من جانب الخادم (SSR) وتحسين الأداء.
واجهة الخلفية (Backend)	مع Python FastAPI	Python هو الخيار الأمثل لمعالجة الذكاء الاصطناعي (AI/ML) نظراً لنوفر مكتبات مثل TensorFlow، PyTorch و FastAPI. FastAPI يوفّر أداءً عالياً جداً (أسرع من Flask و Django) وهو مثالٍ لخدمة استدعاءات API الثقيلة لمعالجة الصور.
قواعد البيانات (Database)	PostgreSQL (SQL)	موثوقية عالية، تكامل بيانات، وقدرات متقدمة مثل تخزين JSONB لإدارة بيانات الصور والمفاتيح (API Keys) بشكل منظم.
تخزين الملفات (File Storage)	Amazon S3 أو Google Cloud Storage	ضروري لتخزين الصور الأصلية والصور المعالجة. يوفّر قابلية توسيع لنهائية وتكلفة فعالة.
إدارة المهام (Task Queue)	Redis مع Celery	حاسّم لمعالجة الصور بشكل غير متزامن (Asynchronous). يضمن أن المستخدم لا ينتظر نتيجة المعالجة، بل يتم إخطاره عند الانتهاء.

2. إدارة المهام غير المتزامنة (Management)

تعتبر معالجة الصور بالذكاء الاصطناعي عملية تستغرق وقتاً طويلاً (قد تصل إلى عدة دقائق). لذلك، فإن استخدام نظام قائمة انتظار غير متزامن هو أمر حاسم لضمان تجربة مستخدم سلسة وموثوقية عالية.

أ. آلية العمل المقترحة (Celery/Redis)

1. **الطلب (Request):** يرفع المستخدم الصورة عبر واجهة الأمامية.
2. **الواجهة الخلفية (FastAPI):** تستقبل الطلب، وتحقق من صلاحية المستخدم، ثم ترسل مهمة المعالجة إلى قائمة انتظار Celery.
3. **الاستجابة الفورية (Immediate Response):** ترسل FastAPI استجابة فورية للمستخدم HTTP (202 Accepted) مع رقم تعريف المهمة (Job ID).
4. **المعالجة (Processing):** تقوم وحدات العمل (Celery Workers) بتنفيذ مهمة الذكاء الاصطناعي على خوادم GPU.

5. الإخطار (Notification): عند الانتهاء، يتم تخزين الصورة الناتجة في S3، ويتم تحديث حالة المهمة في قاعدة البيانات، وإرسال إخطار للمستخدم (عبر WebSockets أو Polling) بأن الصورة جاهزة للتحميل.

ب. الفوائد التقنية

- تجربة مستخدم محسنة: لا يضطر المستخدم لانتظار على صفحة واحدة.
- قابلية التوسيع: يمكن إضافة المزيد من وحدات العمل (Celery Workers) بسهولة لتلبية الطلب المتزايد (Auto-Scaling).
- الموثوقية: يضمن Celery أن المهام التي تفشل يمكن إعادة محاولتها تلقائياً (Automatic Retry) دون تدخل يدوي.

3. متطلبات البنية التحتية للذكاء الاصطناعي (AI Infrastructure) (Requirements)

يتطلب تشغيل نماذج الانتشار (Diffusion Models) عالية الدقة (مثل SDXL) ووحدات معالجة رسوميات (GPU) متخصصة.

التبرير	التصويبة	المطلب
هذه الوحدات هي المعيار الصناعي لعمليات الاستدلال (Inference) والتدريب الدقيق (Fine-Tuning) لنماذج الذكاء الاصطناعي الكبيرة، وتتوفر أداءً عالياً مقابل التكلفة.	H100 أو NVIDIA A100	وحدات المعالجة
يجب إعداد نظام التوسيع التلقائي (Kubernetes أو ما يعادله) لزيادة أو تقليل عدد وحدات GPU العاملة بناءً على حجم قائمة انتظار Celery، لضمان الكفاءة في التكلفة والسرعة في الخدمة.	التوسيع التلقائي (Scaling)	التوسيع
استخدام حاويات Docker لضمان بيئة تشغيل متسقة وموثوقة لجميع مكونات الذكاء الاصطناعي، مما يسهل عملية النشر والإدارة (Deployment).	Docker/Containerization	البيئة