

warning "Mise à jour du backend (octobre 2026)" L'inférence GitHub Models a été retirée le 30 juillet 2026. Les références ci-dessous sont conservées uniquement comme contexte historique pendant la migration de ce lab. Utilisez le Lab 013 ? Backends LLM gratuits pour configurer Foundry Local ou Ollama avant de continuer.

Lab 044 : Phi-4 + Ollama en production

Niveau : L400 Parcours : Pro Code Durée : ~90 min ? Coût : Gratuit (local) ? 8 Go+ de RAM pour Phi-4-mini

Ce que vous apprendrez

Exécuter Microsoft Phi-4 localement via Ollama (coût nul, pas d'internet requis) Évaluer les performances de Phi-4-mini vs GPT-4o-mini sur des tâches de raisonnement Construire un agent local-first avec basculement automatique vers le cloud Ajouter la mise en cache des réponses pour éviter les appels LLM redondants Traitement par lots pour les cas d'utilisation à haut débit

Introduction

Phi-4 est la famille de petits modèles de langage de pointe de Microsoft. Phi-4-mini (3,8 milliards de paramètres) fonctionne sur un ordinateur portable avec 8 Go de RAM et égale GPT-4o-mini sur de nombreux benchmarks de raisonnement ? gratuitement, localement, sans aucun souci de confidentialité.

Ce lab construit une configuration d'inférence locale de qualité production : mise en cache, traitement par lots, basculement et évaluation des performances.

Prérequis

Ollama installé 8 Go+ de RAM (16 Go recommandés pour de meilleures performances) Python 3.11+ pip install openai httpx Optionnel : GITHUB_TOKEN pour le basculement vers le cloud

Exercice pratique

Étape 1 : Télécharger les modèles Phi-4

[diagram or code sample]

Étape 2 : Tester Phi-4 via l'API compatible OpenAI d'Ollama

Ollama expose une API compatible OpenAI à <http://localhost:11434/v1>, vous pouvez donc utiliser le même client Python openai :

[diagram or code sample]

Étape 3 : Construire un client local-first avec basculement vers le cloud

[diagram or code sample]

Étape 4 : Benchmark Phi-4-mini vs GPT-4o-mini

[diagram or code sample]

[diagram or code sample]

Étape 5 : Traitement par lots pour un haut débit

[diagram or code sample]

Comparaison des performances (résultats typiques)

Modèle		Latence moy.		Qualité		Coût		Confidentialité		phi4-mini (local)		800?2000ms	
?????		Gratuit		? Local		phi4 (local)		3000?8000ms		?????		Gratuit	
? Local		gpt-4o-mini (cloud)		300?800ms		?????		~0,00015 \$/1k tok		?? Cloud			

tip "Quand choisir le local" Utilisez Phi-4-mini pour : les tâches à fort volume, les données sensibles, les scénarios hors ligne et le développement. Utilisez le cloud pour : le raisonnement complexe, les SLA de production et les tâches nécessitant les dernières capacités.

Prochaines étapes

Conteneurisez votre configuration IA locale : Ajoutez Ollama au devcontainer (déjà préconfiguré dans .devcontainer/devcontainer.json) Intégrez Phi-4 dans du RAG : ? Lab 022 ?

RAG avec pgvector

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

[Télécharger le PDF généré](#) [Télécharger le ZIP généré](#)

Supporting file: CHEATSHEET.md

Fichiers d'accompagnement: Lab 044 : Phi-4 + Ollama en production

Commencez par le parcours local ou gratuit documenté. Consignez les hypothèses et les preuves de validation. Gardez les services hébergés facultatifs sauf exigence explicite du lab.

Supporting file: README.md

Lab 044 : Phi-4 + Ollama en production

Utilisez ces fichiers avec le guide du lab et le kit généré.