

warning "Backend update (October 2026)" GitHub Models inference was retired on July 30, 2026. Any references below are retained only for historical context while this lab is migrated. Use Lab 013 ? Free LLM backends to configure Foundry Local or Ollama before continuing.

Lab 044: Phi-4 + Ollama in Production

Level: L400 Path: Pro Code Time: ~90 min ? Cost: Free (local) ? 8GB+ RAM for Phi-4-mini
What You'll Learn

Run Microsoft Phi-4 locally via Ollama (zero cost, no internet required) Benchmark
Phi-4-mini vs GPT-4o-mini on reasoning tasks Build a local-first agent with automatic cloud
fallback Add response caching to avoid redundant LLM calls Batch processing for
high-throughput use cases

Introduction

Phi-4 is Microsoft's state-of-the-art small language model family. Phi-4-mini (3.8B parameters) runs on a laptop with 8GB RAM and matches GPT-4o-mini on many reasoning benchmarks ? for free, locally, with no privacy concerns.

This lab builds a production-grade local inference setup: caching, batching, fallback, and benchmarking.

Prerequisites

Ollama installed 8GB+ RAM (16GB recommended for best performance) Python 3.11+ pip install
openai httpx Optional: GITHUB_TOKEN for cloud fallback

Lab Exercise

Step 1: Pull Phi-4 models

[diagram or code sample]

Step 2: Test Phi-4 via Ollama's OpenAI-compatible API

Ollama exposes an OpenAI-compatible API at <http://localhost:11434/v1>, so you can use the same openai Python client:

[diagram or code sample]

Step 3: Build a local-first client with cloud fallback

[diagram or code sample]

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

[diagram or code sample]

[diagram or code sample]

Step 5: Batch processing for high throughput

[diagram or code sample]

Performance Comparison (typical results)

Model	Avg Latency	Quality	Cost	Privacy	phi4-mini (local)	800?2000ms	?????
Free	? Local	phi4 (local)	3000?8000ms	?????	Free	? Local	gpt-4o-mini (cloud)
	300?800ms	?????	~\$0.00015/1k tok	?? Cloud			

tip "When to choose local" Use Phi-4-mini for: high-volume tasks, sensitive data, offline scenarios, and development. Use cloud for: complex reasoning, production SLAs, and tasks requiring latest capabilities.

Next Steps

Containerize your local AI setup: Add Ollama to the devcontainer (already pre-configured in .devcontainer/devcontainer.json) Embed Phi-4 in RAG: ? Lab 022 ? RAG with pgvector

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

Download generated PDF Download generated ZIP

Supporting file: CHEATSHEET.md

Supporting files: Lab 044: Phi-4 + Ollama in Production

Start with the documented local or free path. Record assumptions and validation evidence.

Keep hosted services optional unless the lab marks them required.

Supporting file: README.md

Lab 044: Phi-4 + Ollama in Production

Use these files with the lab guide and generated kit.