PREMIUM GUIDE · LEARN.AIWITHPRIYABRAT.COM

LLMOps: The Production Guide

Take GenAI apps from demo to shipped product: deployment, serving open models, caching, observability, evals in CI, cost control and security, explained step by step with complete code.

By **Priyabrat Dalbehera** · GenAI engineer

9 chapters | **40+** code samples | **30+** model interview answers | **1** launch checklist

FREE PREVIEW · FULL GUIDE AT LEARN.AIWITHPRIYABRAT.COM

01 From demo to production

02 Docker and deployment

03 Serving open models

04 Caching and rate limits

05 Observability with Langfuse

06 Running evals in CI

07 Cost control

08 Security

09 Launch checklist and interview bank

© 2026 Priyabrat Dalbehera

learn.aiwithpriyabrat.com · Edition 1.0 · October 2026

Contents

1 **From demo to production**
What changes in production, the stack, the four questions

2 **Docker and deployment**
Dockerfile, secrets, health checks, Compose, platforms, releases

3 **Serving open models**
API vs self-host, GPU memory, vLLM, Ollama, operations

4 **Caching and rate limits**
Safe cache keys, semantic-cache trap, prompt caching, token limits

5 **Observability with Langfuse**
Traces, what to record, Langfuse setup, feedback, alerts

6 **Running evals in CI**
Eval runner, release gate, GitHub Actions, stable and cheap evals

7 **Cost control**
Attribution, budgets, the savings levers worked through

FREE PREVIEW · FULL GUIDE AT LEARN.AIWITHPRIYABRAT.COM

8 **Security**
OWASP LLM Top 10, secrets, redaction, output handling, abuse

9 **Launch checklist and interview bank**
Go-live checklist, first-week runbook, more answers

Read chapter 1 first
It maps the stack; then jump to the chapter you need.

Run the code
Python samples were run while writing and their outputs are shown; deployment and SDK snippets need their services.

Practise answers
Each chapter ends with model interview answers. Say yours first.

learn.aiwithpriyabrat.com

CHAPTER 1

From demo to production

The gap between a working demo and a shipped product, and the map for the rest of this guide.

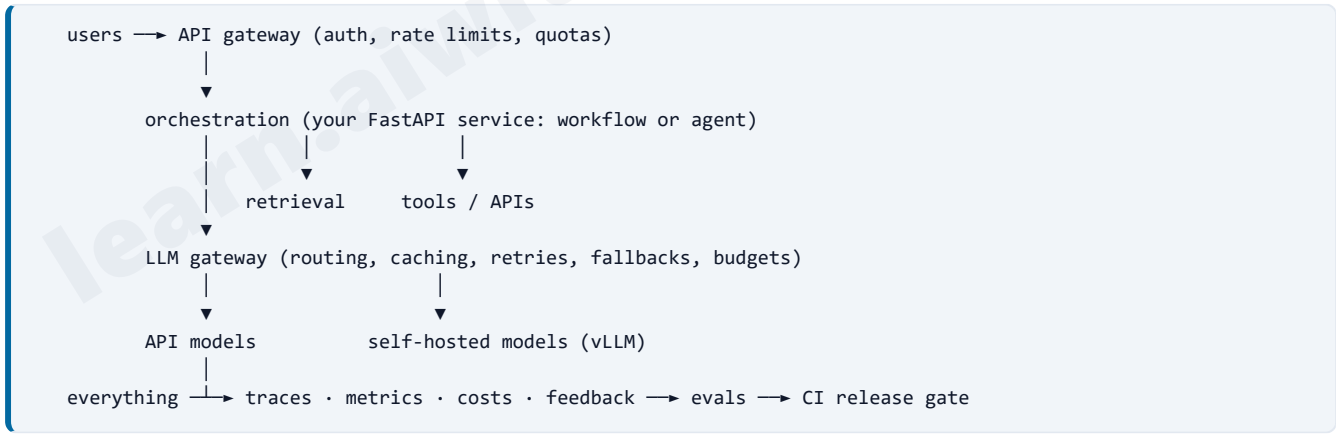
1.1 Why "it works on my laptop" isn't shipped

A GenAI prototype proves that a model *can* do a task. A product has to do it **for thousands of people, every day, at a predictable cost, without leaking data, and without getting quietly worse** when someone changes a prompt. That gap is what this guide is about. It's also the gap interviewers probe with one question: "*and how would you run this in production?*"

In the demo	In production
One user: you	Thousands of users at once, with peaks
You read every answer	Nobody reads most answers; you need measurements
API key in a notebook	Secrets in a vault; separate keys per environment
Cost is a few dollars	Cost is a monthly line item someone asks about
The provider never fails	Rate limits, timeouts and outages happen weekly
Prompt edits are free	A prompt edit can break ten other answers
Data is your test file	Personal data, permissions, retention rules

FREE PREVIEW · FULL GUIDE AT LEARN.AIWITHPRIYABRAT.COM

1.2 The production stack in one picture



Each box is a chapter in this guide:

Layer	What it gives you	Chapter
Packaging and deployment	Same build everywhere, safe releases	2
Model serving	Your own models when APIs don't fit	3
Caching and rate limits	Lower cost, protection from abuse and quotas	4
Observability	See every request; debug any answer	5
Evals in CI	Changes can't silently make things worse	6
Cost control	Predictable bills, the biggest savings	7
Security	Secrets, data, output handling, abuse	8
Launch checklist	One page to review before go-live	9

1.3 The four questions every production LLM system must answer

1. **Is it good?** Quality is statistical. You need an evaluation set, automatic checks, and production signals (feedback, sampled scores). Without them, "it seems fine" is the only answer you have.
2. **Is it fast enough?** LLM calls take seconds. Users judge speed by **time to first token**, so you stream, and you move long work to background jobs.
3. **What does it cost?** Every request is billed by the token. You must know the cost per request, per user and per feature, and have limits before the invoice arrives.
4. **Can we change it safely?** Models, prompts and data all change. You need tests, versioning and rollbacks, and your code, not in the prompt.

FREE PREVIEW · FULL GUIDE AT LEARN.AIWITHPRIYABRAT.COM

Keep these four in mind; every chapter adds tools for answering one or more of them.

1.4 A reference service

Examples in this guide refer to one realistic service: a **support assistant** built with FastAPI. It answers policy questions with RAG, looks up orders through tools, streams answers, and stores sessions and caches in Redis. You'll package it (chapter 2), point it at a self-hosted model (3), put caches and limits in front of it (4), trace it (5), test it in CI (6), budget it (7) and lock it down (8).

> **How to use this guide**
>
> Read chapter 1, then jump to the chapter for the problem you have today. Every chapter ends with **model interview answers**: read the question first, answer it out loud, then compare.

1.5 Interview answers

> **Q What changes when you take an LLM prototype to production?**
>
> Almost everything around the model. You need packaging and safe deployment, an LLM gateway with retries, fallbacks and caching, rate limits and budgets, observability (traces with prompts, tokens, cost and latency per step), an evaluation suite that gates releases, cost attribution, and security controls for secrets, personal data, model output and abuse. The model call itself is the smallest part of the system.

Q How do you know an LLM application is working well in production?

Combine offline and online signals. Offline: an eval suite run on every change. Online: user feedback, sampled LLM-judged scores on live traces (faithfulness, relevance), task-level metrics (resolution rate, "I don't know" rate), plus latency, error and cost dashboards with alerts. Bad production answers become new test cases.

FREE PREVIEW · FULL GUIDE AT LEARN.AIWITHPRIYABRAT.COM

learn.aiwithpriyabrat.com