

Observability & Incident Response for LLM Apps

LangSmith + Langfuse

Course Overview

As LLM-powered applications move from prototypes to production, teams quickly discover that traditional monitoring tools are not designed for the unique challenges of language model systems. Prompt chains, token costs, hallucination patterns, and silent quality degradation all require purpose-built observability. This course teaches developers how to build production-grade observability and incident response workflows for LLM applications using both LangSmith and Langfuse.

Across ten hands-on chapters, learners progress from understanding why LLM observability is fundamentally different, through setting up dual tracing platforms, designing instrumentation schemas, building metrics dashboards, mastering trace-based debugging, and running structured incident response with postmortems. Every chapter includes a Jupyter notebook with real API calls, ensuring that learners build practical skills they can apply immediately in their own projects.

Whether you are a developer building RAG applications, an ML engineer deploying models to production, or a tech lead responsible for LLM reliability, this course provides the tools, patterns, and workflows you need to make your LLM systems observable, debuggable, and incident-ready.

Course Objectives

By the end of this course, learners will be able to:

- Instrument LLM applications with both LangSmith and Langfuse for full trace visibility across prompts, retrievals, and tool calls
- Design production-grade instrumentation with PII redaction, sampling strategies, metadata tagging, and observability contracts

- Build Prometheus-based dashboards tracking p95 latency, failure rates, cost per request, and quality proxy metrics
- Debug production issues using trace replay, root-cause analysis, side-by-side comparison, and bulk regression testing
- Run structured incident response with severity rubrics, circuit breakers, runbooks, and postmortem workflows
- Compare LangSmith and Langfuse across data models, features, and team profiles to choose the right tool for their stack
- Integrate all observability skills into a complete production stack through a final hands-on lab

What You Will Learn

- Why traditional APM tools fail for LLM systems and what LLM-specific observability objects look like
- How to set up and configure LangSmith and Langfuse projects for development and production environments
- How to create manual and automatic traces, add metadata and tags, and query traces programmatically
- How to design trace schemas, redact sensitive data, implement sampling, and write observability contracts for your team
- How to define and collect the five core LLM metrics: p95 latency, failure rate, cost, quality proxy, and release comparison
- How to build structured trace records, persist them as JSON, replay with overrides, and run side-by-side comparisons for debugging
- How to classify LLM-specific incidents, implement circuit breakers, run simulated incident drills, and write postmortems from trace data
- How to evaluate both platforms side by side and design hybrid architectures for teams that need the strengths of each

Prerequisites

- Basic Python programming experience (functions, classes, dictionaries)
- Familiarity with REST APIs and web frameworks (FastAPI experience is helpful but not required)
- A working understanding of what LLMs and RAG systems do (no deep ML theory required)
- An OpenAI API key (for making real LLM calls during the labs)
- Comfort working with Jupyter notebooks and the command line

Who This Course Is For

- Developers building LLM-powered applications who need production-grade reliability and visibility
- ML and AI engineers deploying language models to production and managing their lifecycle
- DevOps and SRE teams responsible for monitoring, alerting, and incident response for LLM services
- Tech leads and engineering managers who need to establish observability standards and incident workflows for their teams
- Backend developers integrating LLM features into existing products who want to ensure operational readiness
- Students and learners who want to go beyond demo-level LLM projects and understand production engineering practices

Course Syllabus

Chapter 1: Why LLM Observability Is Different (~30 minutes)

- Production failure demo showing silent quality degradation
- Why traditional logging and APM tools fall short for LLM systems
- LLM-specific observability objects: traces, spans, generations
- Introduction to LangSmith and Langfuse concepts
- Defining north-star metrics for LLM applications
- Course architecture overview and learning path

Chapter 2: Environment Setup and Baseline App (25–30 minutes)

- Environment verification and dependency setup
- FastAPI RAG app architecture and project layout
- App endpoints and the query-retrieve-generate pipeline
- Test dataset creation with diverse question categories
- Baseline performance metrics: latency, success rate, response length
- Identifying the observability gap in baseline metrics

Chapter 3: LangSmith Setup and First Traces (35–40 minutes)

- LangSmith project setup and configuration
- Automatic tracing with LangChain integration
- Manual tracing using RunTree and the `@traceable` decorator
- Programmatic trace exploration and querying
- Adding metadata and tags for trace filtering
- Debugging a slow request using trace data

Chapter 4: Langfuse Setup and First Traces (40–45 minutes)

- Langfuse project setup and configuration
- Langfuse data model deep dive: traces, spans, generations, events
- Manual trace creation and RAG app instrumentation
- Sessions and conversation grouping
- Custom trace IDs for cross-system correlation
- API-based trace querying and initial platform comparison

Chapter 5: Instrumentation Design for Production (30–35 minutes)

- Defining a trace schema with required fields using Pydantic
- Redacting sensitive data (PII, API keys) before traces leave the app
- Standardizing tags and metadata across both platforms
- Implementing sampling strategies: 100% in dev, 10% + errors in prod
- Writing an observability contract as a team-level agreement
- Wiring everything into a single instrumented query function

Chapter 6: Metrics, Dashboards, and Alerts (30–35 minutes)

- The five core LLM metrics: p95 latency, failure rate, cost, quality proxy, releases
- Building a MetricsCollector that processes trace batches
- Prometheus metrics instrumentation for FastAPI
- Dashboard panel design: latency, failures, releases, quality
- Alert rules and severity levels: WARN vs CRITICAL thresholds

- Alert tuning and release comparison for regression detection

Chapter 7: Trace-Based Debugging and Replay (35–40 minutes)

- Building structured TraceRecord objects as debugging snapshots
- Persisting and loading traces as JSON files
- Root-cause analysis from production traces
- Replaying traces with overrides: prompt version, model, parameters
- Side-by-side trace comparison with structured diffs
- Bulk regression testing across an entire trace library
- The 8-step debug checklist for LLM production issues

Chapter 8: Incident Response for LLM Apps (35–40 minutes)

- LLM-specific incident classes and failure patterns
- Severity classification: SEV-1, SEV-2, and SEV-3 rubrics
- Circuit breaker controls: disabling tools, caching, fallback models
- Simulated incident drills with end-to-end walkthroughs
- Writing postmortems from trace data within 48 hours
- Converting incidents into regression tests for continuous protection

Chapter 9: LangSmith vs Langfuse in Real Teams (15–20 minutes)

- Data model differences: run-centric vs observation-centric approaches
- Feature comparison across tracing, evaluation, cost tracking, deployment, and pricing
- Team profile mapping: which tool fits which team
- Hybrid architecture: using both tools together effectively
- Phased rollout plan from development to production

Chapter 10: Final Lab and Packaging (20–25 minutes)

- Project structure review and integration check
- Full incident lifecycle demo: healthy trace through failure, alert, diagnosis, mitigation, and postmortem
- Course completion checklist export and self-assessment

- Next steps, resources, and recommended learning paths

Skills You Will Gain

- LLM application instrumentation and tracing with LangSmith and Langfuse
- Production observability design: trace schemas, PII redaction, sampling, and observability contracts
- Metrics engineering: defining, collecting, and dashboarding LLM-specific metrics with Prometheus
- Trace-based debugging: replay, comparison, root-cause analysis, and regression testing
- Incident response: severity classification, circuit breakers, runbooks, drills, and postmortems
- Platform evaluation: comparing observability tools and designing hybrid architectures for real teams
- End-to-end observability stack integration for production LLM applications