Observability vs monitoring: what is the difference?

1 November 2026 · 5 min read · blogs.iictp.org/observability-vs-monitoring

Monitoring checks for problems you already know about, using predefined measurements and alerts. Observability is the ability to ask new questions about a running system from the data it emits, which matters most when something fails in a way nobody predicted.

TL;DR The short version

- Monitoring watches known signals and alerts on thresholds; observability lets you ask new questions when something fails in an unpredicted way.
- You need both, built on structured logs, traces that carry a request ID, and metrics for rates, errors and durations.
- Alert on what users feel, and start by instrumenting your most important user journey end to end with a standard such as OpenTelemetry.

Read online, listen to it, or change the text size: https://blogs.iictp.org/observability-vs-monitoring

What does monitoring do?

It watches known signals, such as error rate or disk space, and alerts when they cross a threshold. It answers the question “is it broken?”, quickly and cheaply, for the failures you thought of in advance.

What does observability add?

It adds enough detail, through traces, metrics and logs that can be correlated, to answer “why is it broken, for whom, and since when?”. A single request can be followed across services without having planned for that exact question.

Do you need both?

Yes. Monitoring tells you quickly that something is wrong, and observability helps you find out what. A system with alerts and no detail is fast to notice and slow to fix.

What makes a system observable?

Structured logs with consistent fields, traces that carry a request ID across service boundaries, and metrics for rates, errors and durations. Context such as version, region and customer tier attached to the data makes it far easier to slice.

What should you alert on?

Alert on what users feel, such as failed requests or slow responses, rather than on every internal cause such as CPU. Cause-based alerts are noisy, and a team that ignores noisy pages will miss the real one.

Where should you start?

Pick your most important user journey and instrument it end to end with a standard such as OpenTelemetry. Then add detail where incidents show you were blind.

> Related credential: Clean Software Telemetry Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. OpenTelemetry. Observability primer. opentelemetry.io/docs/concepts/observability-primer/.
2. Google. Site Reliability Engineering, “Monitoring Distributed Systems”. sre.google/sre-book/monitoring-distributed-systems/.

Published by IICTP, an independent technical credential body, on 1 November 2026. The current version of this post is at https://blogs.iictp.org/observability-vs-monitoring. More guides: blogs.iictp.org.