DevOpsSchool

Course curriculum · November 2026

LEVEL 04 OF 04 · ADVANCED

Obserbability Advanced Training

Scale an observability platform for many teams with long-term storage, sampling, cost controls and organisation-wide standards.

LIVE ONLINE **Max 8-10 hours**	CORPORATE (DAY-WISE) **5 days · Days 1-5**	SELF-PACED VIDEO **1-year LMS access**
BATCH SIZE **Max 10 learners**	NEXT BATCHES **1st of every month · weekends: first Saturday**	CERTIFICATE **Verifiable certificate of completion**

Foundation	Essential	Intermediate	Advanced
1 day · Included	2 days · Included	3 days · Included	5 days · This curriculum

INSIDE THIS CURRICULUM

What you will achieve	Outcomes and level focus
Who should attend	Audience and prerequisites
Agenda	49 topics · Foundation to Advanced
Hands-on labs	4 labs
Formats and fees	Live, 1:1, self-paced, corporate
Trainer and reviews	Rajesh Kumar
Enrolment	Batches, payment, contacts

WHY DEVOPSSCHOOL

- Rated 4.6 on Google
- 1,500+ verifiable certificates issued in the public registry
- Trainer Rajesh Kumar: 20 years in industry
- Small batches: max 10 learners, every session recorded

Course page: www.devopsschool.com/courses/observability/obserbability-advance-training.html
Talk to us: WhatsApp +91 99057 40781 · +1 (469) 756-6329 · contact@DevOpsSchool.com · reply within 1 hour

OUTCOMES

What you will achieve

By the end of the Obserbability course path, you will be able to:

1

Understand the three signals

Explain how logs, metrics and traces answer different questions and how they correlate.

2

Instrument and observe services

Instrument applications with OpenTelemetry and build dashboards and SLO alerts teams can act on.

3

Run observability at scale

Operate a scalable, cost-controlled observability platform with clear standards for every team.

WHAT THE ADVANCED LEVEL ADDS

Scale, cost, reliability and observability governance

Hands-on: Scale an observability platform. Deploy Mimir, Loki and Tempo on object storage, apply tail sampling and retention policies, and produce a telemetry cost report.

WHO SHOULD ATTEND

- Experienced Obserbability practitioners responsible for quality, security and scale.
- Senior engineers, architects and leads who design and troubleshoot Obserbability solutions.
- Teams making Obserbability an organisation-wide standard.

BEFORE YOU START

- The Intermediate level, or equivalent hands-on experience.
- You can already: Define SLOs, alert on error budget burn, deploy the stack on Kubernetes as code and troubleshoot incidents from alert to root cause.
- Basic computer skills and a laptop with a stable internet connection for the live sessions and labs.
- This curriculum includes every earlier level, so it is complete on its own.

AGENDA · FOUNDATION TO ADVANCED

Obserbability Advanced Training agenda

49 topics. Each level builds on the one before it, so this curriculum lists every earlier level in full. Days are for corporate (day-wise) delivery; live online batches cover the same agenda in max 8-10 hours.

DAY 1 — Foundation: Observability concepts: logs, metrics, traces, alerting

Included from Foundation

1. What is observability: monitoring versus observability, known unknowns and unknown unknowns
2. Why observability is important: faster detection and recovery (MTTD, MTTR) in cloud-native systems
3. Key concepts and terminology: telemetry, signals, cardinality, golden signals, SLIs and SLOs
4. Logging overview and types of logs: application, system, access, audit and security; structured vs plain text
5. Log collection, aggregation, analysis and visualization: agents, shippers, central stores and log queries
6. Metrics overview and types of metrics: counters, gauges, histograms and summaries; RED and USE methods
7. Metric collection, storage, analysis and visualization: pull vs push, time-series databases and dashboards
8. Tracing overview and types of traces: distributed traces, spans, context propagation, client and server spans
9. Trace collection, storage, analysis and visualization: collectors, trace backends and waterfall views
10. Alerting overview: symptom versus cause alerts, thresholds, routing, on-call and alert fatigue
11. Best practices for implementing observability: instrumentation standards, naming and retention
12. Popular observability tools and frameworks: OpenTelemetry, Prometheus, Grafana, Loki, Jaeger, Elastic and APM suites
13. Case studies: examples of successful observability implementations, with Q&A and discussion

DAY 2 — Essential: OpenTelemetry instrumentation and Grafana stack setup

Included from Essential

14. OpenTelemetry architecture: API, SDK, OTLP protocol, Collector and semantic conventions
15. Auto-instrumentation: the Java agent, Python opentelemetry-instrument, and Node.js and .NET options
16. Manual instrumentation: creating spans, attributes, events and custom metrics in code
17. OpenTelemetry Collector: receivers, processors, exporters and pipelines in the contrib distribution
18. Prometheus metrics: scrape configs, node_exporter and PromQL rate, sum by and histogram_quantile
19. Grafana Loki logs: the Grafana Alloy agent, labels, LogQL queries and line filters
20. Grafana Tempo traces: OTLP ingestion, TraceQL search and service graphs
21. Grafana dashboards: data sources, panels, variables and dashboard JSON kept in Git
22. Correlating signals: trace IDs in logs, exemplars and jumping from metrics to traces to logs
23. Kubernetes telemetry: kube-state-metrics, cAdvisor, Kubernetes events and pod logs
24. On-call dashboards: RED and USE views per service and avoiding vanity panels
25. Team conventions: service.name and resource attributes, ownership labels and dashboard reviews

DAY 3 — Intermediate: Alerting, SLOs, automation and troubleshooting

Included from Intermediate

26. Alerting with Alertmanager and Grafana Alerting: rules, labels, routing trees, silences and inhibition
27. SLIs and SLOs in practice: availability and latency SLIs, error budgets and multi-window burn-rate alerts
28. Incident response: on-call rotations, escalation, runbooks linked from alerts and PagerDuty integration
29. Collector processing: batch, memory_limiter, attributes, filter and redaction processors
30. Kubernetes deployment: OpenTelemetry Operator, DaemonSet and gateway Collectors, kube-prometheus-stack

31. Observability as code: Grafana provisioning, the Terraform Grafana provider and PrometheusRule resources
32. Troubleshooting workflow: from alert to dashboard to trace to log, isolating slow dependencies
33. Continuous profiling: Grafana Pyroscope and eBPF-based CPU and memory profiles
34. Synthetic and real user monitoring: blackbox_exporter probes, Grafana k6 checks and Faro browser telemetry
35. Security basics: PII redaction, TLS for OTLP and access control on dashboards and data sources
36. Middleware telemetry: exporters for MySQL, PostgreSQL, Redis and Kafka, and query latency insights
37. Telemetry quality: cardinality control, label hygiene and dropping noisy logs and metrics

DAYS 4-5 **Advanced: Scale, cost, reliability and observability governance** — New at this level

38. Scaling metrics: Grafana Mimir or Thanos for long-term, highly available Prometheus storage
39. Scaling logs and traces: Loki and Tempo in microservices mode with object storage and retention
40. High availability: Prometheus HA pairs, Alertmanager clustering and the Collector load-balancing exporter
41. Sampling strategy: head versus tail sampling, probability sampling and its cost impact
42. Cost management: telemetry volume budgets, adaptive metrics, tiered retention and cost per signal
43. eBPF observability: zero-code instrumentation with Grafana Beyla or OpenTelemetry eBPF, and Cilium Hubble
44. AIOps and anomaly detection: baselines, outlier detection, alert correlation and noise reduction
45. Multi-tenant observability: tenant isolation, per-team limits and federated views for platform teams
46. Governance: organisation standards for semantic conventions, instrumentation libraries and review gates
47. Vendor strategy: OpenTelemetry to avoid lock-in, routing to several backends and agent migrations
48. Reliability testing: verifying alerts and dashboards with fault injection and game days
49. Capstone: an end-to-end observability platform for a microservices app with SLOs and runbooks

PRACTICE

Hands-on labs

You build these yourself during the course, guided by the trainer.

FOUNDATION **Explore logs, metrics and traces**
Run a small demo application and use its logs, metrics and traces side by side to answer a real troubleshooting question.

ESSENTIAL **Instrument an app with OpenTelemetry**
Instrument a multi-service app with OpenTelemetry, route telemetry through a Collector to Prometheus, Loki and Tempo, and build a correlated Grafana dashboard.

INTERMEDIATE **Build SLO alerts and runbooks**
Define SLOs for a service, create burn-rate alerts routed through Alertmanager and resolve an injected latency fault using traces and logs.

ADVANCED **Scale an observability platform**
Deploy Mimir, Loki and Tempo on object storage, apply tail sampling and retention policies, and produce a telemetry cost report.

HOW YOU CAN LEARN

Formats and fees

The same agenda in four formats. Fees are in Indian rupees as of November 2026; the course page shows them in your currency.

FORMAT	DURATION	WHAT YOU GET	FEE
Live & interactive (online)	Max 8-10 hours	Instructor-led public batch of max 10 learners with demos, labs, Q&A and assignment review. Weekday batches start on the 1st of every month, weekend batches on the first Saturday of every month.	**INR 34,999**
1:1 live training (online)	Max 8-10 hours, one-to-one	The same agenda, taught one-to-one by the trainer. Starts immediately, without waiting for the next batch; sessions planned around your working hours, at your pace.	**INR 99,999**
Self-paced video	1-year LMS access	All four Obserbability levels, plus 20+ more courses and 50+ tools, at your own pace.	**INR 833/month** billed yearly at INR 9,996
Corporate (day-wise)	5 days for this level	A private batch for your team, on demand. Foundation 1 day, Essential 2, Intermediate 3, Advanced 5 (cumulative).	**Quote on request**

ASSESSMENT AND CERTIFICATE

Your progress is checked through hands-on labs and assignments at every level. On completion you receive a DevOpsSchool certificate of completion with a unique credential ID that anyone can verify at devopsschool.com/certificates.

INCLUDED WITH EVERY LIVE BATCH

- Every live session is recorded and stays in your LMS for a year.
- Can't attend your batch? Move to next month's batch.
- Free demo class before you pay.
- Free 10-15 minute call with an advisor or the trainer to check the course fits you.

YOUR TRAINER

Learn from Rajesh Kumar

Principal Architect, trainer and mentor in DevOps, SRE, DevSecOps and platform engineering, with 20 years across 8+ software MNCs, including Staff DevOps/SRE Engineer for a leading Japanese mobile-payment platform and Principal DevOps Architect and Manager in Europe.

Profile: www.rajeshkumar.xyz · Videos: youtube.com/@TheDevOpsSchool

20 years in industry	**200+** corporate trainings
10,000+ engineers mentored	**70+** organisations coached

REVIEWS

What learners say

Rated 4.6 on Google · more reviews at devopsschool.com/testimonials and in our video testimonials.

> “Rajesh is very good trainer. Rajesh was able to resolve our queries and question effectively. We really liked the hands-on examples covered during this training program.”
>
> Indrayani, India

> “The training was very useful and interactive. Rajesh helped develop the confidence of all.”
>
> Abhinav Gupta, Pune

START YOUR OBSERBABILITY TRAINING

Join the next batch

Weekday batches start on the 1st of every month and weekend batches on the first Saturday of every month, with max 10 learners each. 1:1 training can start immediately; corporate batches run on demand. We reply within 1 hour.

Pay by UPI, credit and debit cards, net banking or PayPal. Before you pay, read our refund policy. All levels of this course: www.devopsschool.com/courses/observability/

ENROL OR RESERVE A SEAT
Obserbability Advanced Training

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com