DevOpsSchool

Course curriculum · November 2026

LEVEL 04 OF 04 · ADVANCED

OpenTelemetry Advanced Training

Scale OpenTelemetry pipelines for large estates, build custom Collectors, manage fleets and lead migrations from proprietary agents.

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	47 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/opentelemetry/opentelemetry-and-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 OpenTelemetry course path, you will be able to:

1

Instrument any service

Instrument applications with OpenTelemetry automatically and manually across languages.

2

Build Collector pipelines

Design Collector pipelines that enrich, filter, secure and route telemetry to any backend.

3

Standardise telemetry at scale

Lead OpenTelemetry adoption, governance and migration across large organisations.

WHAT THE ADVANCED LEVEL ADDS

Scale, custom distributions, governance and migration

Hands-on: Design an enterprise telemetry pipeline. Build a custom Collector distribution, manage agents with OpAMP and migrate one service from a proprietary agent while dual-shipping data.

WHO SHOULD ATTEND

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

BEFORE YOU START

- The Intermediate level, or equivalent hands-on experience.
- You can already: Run OpenTelemetry on Kubernetes with the Operator, apply transformation and tail sampling in the Collector and fix broken traces.
- 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

OpenTelemetry Advanced Training agenda

47 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: OpenTelemetry concepts, signals and first instrumentation

Included from Foundation

1. Why OpenTelemetry: vendor-neutral telemetry, the merger of OpenTracing and OpenCensus and CNCF governance
2. Signals: traces, metrics and logs status, plus profiles under development
3. Core components: API, SDK, instrumentation libraries, OTLP, Collector and semantic conventions
4. Traces: spans, span kinds, attributes, events, links, status and the trace and span IDs
5. Context propagation: W3C Trace Context and Baggage headers between services
6. Resources: service.name, service.version, deployment environment and resource detectors
7. Metrics model: counters, up-down counters, histograms, gauges and observable instruments
8. Logs model: the logs bridge, log records with trace context and existing logging libraries
9. Running the Collector: core and contrib distributions, a minimal OTLP receiver and debug exporter
10. First instrumentation: auto-instrument a Python or Java service and export to Jaeger
11. The OpenTelemetry Demo: exploring a reference microservices app and its telemetry

DAY 2 — Essential: SDK configuration, manual instrumentation and Collector pipelines

Included from Essential

12. SDK configuration: tracer, meter and logger providers, exporters and OTEL_ environment variables
13. Auto-instrumentation by language: Java agent, opentelemetry-instrument for Python, Node.js and .NET
14. Manual tracing: creating spans, setting attributes, recording exceptions and span status
15. Custom metrics: choosing instruments, attributes, views and explicit or exponential histogram buckets
16. Logs correlation: injecting trace and span IDs into application logs
17. Semantic conventions: HTTP, database, messaging and RPC attribute names
18. Collector pipelines: receivers, processors and exporters for traces, metrics and logs
19. Essential processors: batch, memory_limiter, resourcedetection and attributes
20. Backends: Jaeger and Tempo for traces, Prometheus for metrics and Loki or OpenSearch for logs
21. Prometheus interoperability: prometheus receiver, remote write exporter and OTLP ingestion in Prometheus
22. Head sampling: AlwaysOn, TraceIdRatioBased and ParentBased sampling strategies
23. Team conventions: span naming, attribute standards and instrumentation code review

DAY 3 — Intermediate: Kubernetes, transformation, tail sampling and troubleshooting

Included from Intermediate

24. OpenTelemetry Operator: OpenTelemetryCollector and Instrumentation resources for auto-injection
25. Deployment patterns: agent DaemonSet, gateway Deployment and sidecar Collectors
26. Kubernetes enrichment: k8sattributes processor, kubeletstats and k8s_cluster receivers
27. Transforming telemetry: the transform processor and OTTL, filter and redaction processors
28. Tail sampling: tail_sampling policies for errors and latency and the load-balancing exporter
29. Connectors: spanmetrics and servicegraph connectors for RED metrics and dependency maps

30. Routing: sending different tenants or signals to different backends with the routing connector
31. eBPF instrumentation: zero-code tracing with OpenTelemetry eBPF Instrumentation for uninstrumented services
32. Security: TLS and authentication extensions for OTLP, PII redaction and secret handling
33. Collector self-monitoring: internal metrics, health_check, zpages and pprof extensions
34. Troubleshooting: missing context, broken propagation, clock skew, dropped spans and exporter retries
35. Testing instrumentation: in-memory exporters and asserting spans in unit tests

DAYS 4-5 **Advanced: Scale, custom distributions, governance and migration** — New at this level

36. Scaling Collectors: horizontal scaling, queues, retries, persistent sending queues and backpressure
37. High availability: redundant gateways, Kafka buffering and multi-region telemetry paths
38. Custom distributions: building a Collector with the OpenTelemetry Collector Builder (ocb)
39. Extending OpenTelemetry: writing custom processors or receivers and custom instrumentation libraries
40. Fleet management: OpAMP for remote configuration and health of thousands of agents
41. Declarative configuration: file-based SDK configuration for consistent settings across languages
42. Cost control: sampling strategy, metric cardinality limits, delta versus cumulative temporality
43. Governance: organisation semantic conventions, schema URLs, ownership and instrumentation standards
44. Migration: moving from vendor agents, OpenTracing or OpenCensus to OpenTelemetry in phases
45. Multi-backend strategy: dual shipping, vendor evaluation and avoiding lock-in
46. Profiling signal: continuous profiling integration and correlating profiles with traces
47. Capstone: an enterprise telemetry pipeline with tail sampling, routing, redaction and SLO dashboards

PRACTICE

Hands-on labs

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

FOUNDATION **Send your first OpenTelemetry traces**
Auto-instrument two small services, send OTLP data through a local Collector and follow one request across both services in Jaeger.

ESSENTIAL **Build a full telemetry pipeline**
Add manual spans and business metrics to a service and route traces, metrics and logs through one Collector to Tempo, Prometheus and Loki.

INTERMEDIATE **Operate OpenTelemetry on Kubernetes**
Deploy the Operator with auto-injected instrumentation, run agent and gateway Collectors with tail sampling and fix a broken propagation path.

ADVANCED **Design an enterprise telemetry pipeline**
Build a custom Collector distribution, manage agents with OpAMP and migrate one service from a proprietary agent while dual-shipping data.

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 OpenTelemetry 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 OPENTELEMETRY 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/opentelemetry/

ENROL OR RESERVE A SEAT
OpenTelemetry Advanced Training

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com