DevOpsSchool

Course curriculum · November 2026

LEVEL 04 OF 04 · ADVANCED

Opentracing Advanced Training

Operate tracing at production scale with tail sampling, resilient pipelines and governance, and complete a phased migration off OpenTracing.

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/opentracing/opentracing-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 Opentracing course path, you will be able to:

1

Understand distributed tracing

Explain spans, context propagation and the OpenTracing data model in real systems.

2

Instrument services correctly

Add consistent, tested tracing to services in several languages and protocols.

3

Modernise tracing at scale

Bridge and migrate to OpenTelemetry and operate governed trace pipelines with smart sampling.

WHAT THE ADVANCED LEVEL ADDS

Scale, sampling strategy, governance and full migration

Hands-on: Capstone: production tracing platform migration. Replace OpenTracing instrumentation in a microservices app with native OpenTelemetry, add tail sampling and resilient Collector tiers, and validate traces in CI.

WHO SHOULD ATTEND

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

BEFORE YOU START

- The Intermediate level, or equivalent hands-on experience.
- You can already: Keep OpenTracing code working on OpenTelemetry through shims, route traces through the Collector, and use traces to find latency and error root causes.
- 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

Opentracing 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: Tracing concepts, OpenTracing data model and Jaeger

Included from Foundation

1. Why distributed tracing: requests crossing many services, latency attribution and limits of logs and metrics alone
2. OpenTracing project history: CNCF origins, the vendor-neutral API idea, the 2019 merger into OpenTelemetry and archival
3. Data model: traces, spans, operation names, start and finish timestamps and span duration
4. Span relationships: SpanContext, ChildOf and FollowsFrom references
5. Tags, logs and baggage: span tags for indexing, timestamped span logs and request-scoped baggage items
6. Tracer interface: StartSpan options, GlobalTracer registration and the no-op tracer
7. Context propagation: Inject and Extract with text map, HTTP headers and binary carriers
8. Jaeger architecture: client, agent, collector, query service and storage, and Jaeger v2 on the OpenTelemetry Collector
9. Jaeger all-in-one: running it in Docker, opening the UI and searching traces by service and tags
10. First trace: instrumenting a hello world service in Go or Python with an OpenTracing-compatible tracer
11. Two-service trace: propagating context over HTTP and viewing the parent and child spans in Jaeger

DAY 2 — Essential: Instrumenting real services in several languages

Included from Essential

12. Java instrumentation: ScopeManager, activating spans, try-with-resources scopes and opentracing-contrib modules
13. Go instrumentation: context.Context propagation, StartSpanFromContext and middleware for net/http
14. Python and Node.js instrumentation: contextvars-based scope managers and Express or Flask middleware
15. gRPC tracing: client and server interceptors and metadata propagation
16. Database and cache spans: JDBC, SQL and Redis calls with db.type, db.statement and peer tags
17. Messaging spans: Kafka or RabbitMQ producers and consumers with FollowsFrom references
18. Standard tag conventions: span.kind, http.method, http.url, http.status_code, component and error
19. Error handling in spans: setting the error tag, logging exceptions with event, message and stack fields
20. Zipkin as a backend: B3 propagation headers, Zipkin server and comparing its UI with Jaeger
21. Sampling configuration: constant, probabilistic and rate-limiting strategies and remote sampling in Jaeger
22. Testing instrumentation: MockTracer assertions on finished spans, tags and references
23. Instrumentation guidelines: span naming, granularity and what to trace first in a service

DAY 3 — Intermediate: OpenTelemetry bridge, pipelines and trace analysis

Included from Intermediate

24. OpenTracing to OpenTelemetry mapping: Tracer, Span, SpanContext, references, tags and logs to OTel concepts
25. OpenTracing shim: running legacy instrumentation on the OpenTelemetry SDK in Java, Go and Python
26. W3C Trace Context: traceparent and tracestate headers and coexisting with B3 and uber-trace-id
27. OpenTelemetry Collector: receivers for Jaeger, Zipkin and OTLP, batch processor and exporters to several backends
28. Auto-instrumentation alongside manual spans: Java agent and zero-code options with existing OpenTracing code

29. Service mesh tracing: Envoy and Istio generated spans and header forwarding requirements in applications
30. Trace analysis: critical path, latency outliers, comparing traces and service dependency graphs
31. Root cause analysis: from a slow endpoint to the misbehaving downstream call using tags and span logs
32. Grafana Tempo and TraceQL: storing traces cheaply and querying spans by attributes and duration
33. Span metrics: deriving request rate, error and duration metrics and service graphs from traces
34. Data hygiene: keeping secrets and personal data out of tags and baggage, attribute redaction in the Collector
35. Troubleshooting broken traces: lost context in thread pools and async code, clock skew and dropped spans

DAYS 4-5 **Advanced: Scale, sampling strategy, governance and full migration** **New at this level**

36. Tail-based sampling: keeping errors and slow traces with Collector policies and load-balancing exporters
37. Pipeline resilience: collector tiers, queues and retries, Kafka buffering and backpressure handling
38. Storage at scale: Jaeger with Elasticsearch, OpenSearch or Cassandra, retention tiers and index sizing
39. Overhead management: measuring instrumentation cost, batching settings and adaptive sampling
40. Event-driven architectures: span links, batch consumers and tracing across message queues and serverless
41. Correlated troubleshooting: trace IDs in logs, exemplars on metrics and linked dashboards
42. Trace-based testing: asserting on spans in integration tests and in CI with tools such as Tracetest
43. eBPF tracing: zero-code spans for services you cannot modify and their limits
44. Governance: instrumentation standards, semantic conventions, ownership and review checklists
45. Compliance: data residency of trace storage, retention rules and audit requirements
46. Phased migration plan: inventory of OpenTracing usage, dual-run validation and removing the shim
47. Capstone: migrate a traced microservices system to native OpenTelemetry with tail sampling and SLO dashboards

PRACTICE

Hands-on labs

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

FOUNDATION **Trace a request across two services**
Run Jaeger all-in-one, instrument a frontend and a backend service with OpenTracing, propagate context over HTTP and inspect the resulting trace timeline.

ESSENTIAL **Instrument a polyglot order system**
Instrument a Java API, a Go worker and a Python service connected by gRPC and Kafka, add database spans and verify the instrumentation with MockTracer tests.

INTERMEDIATE **Bridge legacy tracing to OpenTelemetry**
Run the existing OpenTracing services through the OTel shim and Collector into Jaeger and Tempo, switch to W3C propagation and trace a slow request to its root cause.

ADVANCED **Capstone: production tracing platform migration**
Replace OpenTracing instrumentation in a microservices app with native OpenTelemetry, add tail sampling and resilient Collector tiers, and validate traces in CI.

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 Opentracing 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 OPENTRACING 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/opentracing/

ENROL OR RESERVE A SEAT
Opentracing Advanced Training

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com