joindevops.com

10+ YEARS · 90+ BATCHES · 9,000+ PLACED

• PROGRAM SYLLABUS & BROCHURE

Observability Mastery for DevOps & SRE Engineers

> 4 Weeks. 25 Hours. From "I keep dashboards green" to "I design the systems that make incidents short and rare."

25 hrs	33	1	7
HANDS-ON, 4 WEEKS	MODULE GUIDES	REAL BANKING APP LAB	BONUS MODULES

bankobserve360 — incident triage

```
$ histogram_quantile(0.99, sum by (le) (rate(http_request_duration_seconds_bucket[5m])))
# p99 transfer latency spike detected on payments-svc
⚠ burn-rate alert fired → pivoting to trace
$ trace_id=8f2e1a... → tempo → loki  ✓ correlated in 47s
# root cause: connection-pool exhaustion, postgres-accounts
```

PROGRAM FEE

₹8,990/-

EMI available via Credit Card

joindevops.com · **Live Sessions + Slack Community**

01 The Reality

You can write up{} — but freeze on histogram_quantile in an interview.

You ship code — but can't defend an **SLO target** to leadership.

You get paged — but the alert wakes you up for **causes**, not customer pain.

You see a slow request — but can't pivot from metric → trace → log in under 5 minutes.

You can build a dashboard — but can't design one a **stranger can use at 3 AM**.

Every observability course on the market either teaches **one vendor's UI** or drowns you in **three-pillar theory**. Neither makes you employable at the senior level. Neither solves a real incident. **This one does.**

02 What This Course Is

A 4-week, **25-hour, hands-on bootcamp** that takes you through the entire modern observability stack — built around bankobserve360, a realistic multi-service banking application with the same failure modes you'll meet in production: cache stampedes, connection-pool exhaustion, retry storms, OOMs, slow downstreams, the 3 AM DNS surprise.

01
Instrumented a fleet end-to-end with OpenTelemetry

02
Built a dashboards-as-code Grafana setup

03
Defined a real SLO with multi-window multi-burn-rate alerts

04
Pivoted from a burn-rate alert → trace → log line in under 60 seconds

05
Run a simulated incident as Incident Commander and written the postmortem

06
Left with a portfolio on GitHub you link straight from your resume

03 Is This For You?

Take this course if you

- ✓ Have 6+ months of DevOps / SRE / Platform / Cloud engineering experience
- ✓ Are comfortable with Linux, Docker, basic Kubernetes (kubectl), and Git
- ✓ Have written or maintained a CI/CD pipeline
- ✓ Want to move from "monitoring engineer" to "observability practitioner"
- ✓ Are preparing for senior SRE / Observability / Platform interviews
- ✓ Are tired of using tools instead of picking them

This is NOT for you if

- ✗ You're new to engineering without Linux/Docker fundamentals (take a foundations course first)
- ✗ You want vendor-specific certification (Datadog, New Relic) — this is vendor-neutral by design
- ✗ You want theory without labs — every module here has hands-on work

04 What You'll Be Able To Do

On a system you've never seen before, you will be able to:

01 **Instrument** an HTTP/gRPC service end-to-end with metrics, structured logs, and OpenTelemetry traces — and defend every design choice

02 **Write production-grade PromQL and LogQL** for incident triage and SLO reporting, including correct histogram quantile math

03 **Design an SLO** with a multi-window multi-burn-rate alerting policy aligned to customer impact

04 **Diagnose a live incident** by pivoting from a symptom-based alert through metrics → traces → logs in minutes

05 **Build dashboards-as-code** that survive team turnover

06 **Instrument business-level KPIs** — UPI volume, loan disbursement, fee income, KYC funnel — and build dashboards leadership actually reads

07 **Run a blameless postmortem** that produces follow-through, not theatre

08 **Speak fluently** about trade-offs between Prometheus/Grafana, ELK, OpenTelemetry, Datadog, New Relic, Dynatrace, Splunk, Honeycomb, Chronosphere

05 The Lab Application

bankobserve360

Every lab modifies the same realistic banking system. By the end, you have a fully instrumented, SLO-monitored, alert-routed, postmortem-tested platform you can demo in interviews.

api
Customer-facing API — accounts, transfers, statements

payments-svc
Payments processing with external 3rd-party calls

fraud-scoring
CPU-intensive, latency-sensitive scoring service

notifications
Async, queue-driven notification service

postgres + redis
Accounts DB, plus sessions & rate limiting

load-generator
Produces realistic banking traffic + chaos hooks

06 Course Structure

PHASE	TIME	WHAT YOU BUILD
Pre-course Bootcamp	~4 hrs	A working local Kubernetes lab with bankobserve360 deployed
Week 1 — Foundations + Metrics	~6 hrs	A Service Health dashboard using RED + USE + Golden Signals
Week 2 — Logs + Distributed Tracing	~6.5 hrs	End-to-end OpenTelemetry tracing + structured logging with full correlation
Week 3 — SLOs + Alerting + Correlation	~6 hrs	A production-grade SLO with burn-rate alerts routed to a pager
Week 4 — Advanced + Capstone	~6 hrs	A simulated incident worked end-to-end with a written postmortem
Total	~25 hrs	A portfolio-ready observability platform

07 Week-by-Week Modules

PRE Pre-Course Bootcamp

~4 hrs · self-paced

- **B1** Linux & CLI refresher — the 7 journalctl invocations you'll run every week
- **B2** Networking recap — HTTP status codes as a triage tool, latency vs tail-latency
- **B3** Containers & Kubernetes for observability — OOMKilled vs Evicted, QoS classes
- **B4** YAML, JSON & Regex — the YAML mistakes that bite everyone; PromQL anchoring gotchas
- **B5** The mindset shift — monitoring vs observability, cardinality, symptom-vs-cause alerting
- **B6** Lab setup — a working kind cluster with bankobserve360, verified end-to-end

W1 Foundations + Metrics

~6 hrs

- **1.1** Observability fundamentals — pillars, MELT, push vs pull, why OpenTelemetry is the biggest shift in a decade
- **1.2** Metrics & Prometheus architecture — counter/gauge/histogram/summary, scraping, service discovery
- **1.3** PromQL — instant vs range vectors, rate, histogram_quantile, topk, recording rules
- **1.4** Grafana — dashboards-as-code, variables, exemplars, the seven anti-patterns to reject
- **1.5** RED, USE, Golden Signals + capstone lab: complete Bank Service Health dashboard

W2 Logs + Distributed Tracing

~6.5 hrs

- **2.1** Logging fundamentals — structured JSON, log levels, trace_id injection
- **2.2** Log aggregation — Loki vs ELK architecturally, LogQL filter ordering, retention tiering
- **2.3** Distributed tracing concepts — spans, propagation, head vs tail sampling
- **2.4** OpenTelemetry — API/SDK/Collector/OTLP, auto vs manual instrumentation, semantic conventions
- **2.5** Tempo + Jaeger + capstone lab: trace a slow payment end-to-end and pivot to logs

W3 SLOs, Alerting, Correlation

~6 hrs

3.1	SLI / SLO / SLA / error budgets — Google's multi-window multi-burn-rate alerting recipe
3.2	Alerting & on-call — three properties of good alerts, Alertmanager routing/inhibition/silences
3.3	Correlation across pillars — exemplars, traces vs logs, the 3-click triage goal
3.4	Kubernetes observability deep dive — kube-state-metrics, control plane SLOs
3.5	Synthetic monitoring & RUM — blackbox-exporter, k6, Core Web Vitals
3.6	Capstone lab: complete SLO + burn-rate alerts + dashboard + runbook for the transfer endpoint

W4 Advanced + Production + Capstone

~6 hrs

4.1	Continuous profiling — Pyroscope/Parca, flame graphs, trace vs profile correlation
4.2	eBPF observability — Cilium/Hubble, Pixie, Parca-agent
4.3	Incident response & runbooks — IC role, mitigation vs resolution, blameless postmortems
4.4	Job market wrap-up — commercial APM tour, live-debugging round, system design prompt
4.5	Final capstone: a simulated incident on bankobserve360 — detect, triage, mitigate, postmortem

08 Tools & Technologies Covered

All hands-on unless marked otherwise. This is the full stack you'll operate:

Prometheus	PromQL	Grafana	OpenTelemetry
Loki	LogQL	Tempo	Jaeger
Promtail	Fluent Bit	Alertmanager	kube-prometheus-stack
Thanos / Mimir	Pyroscope / Parca	Cilium / Hubble	Helm + kind
k6 (Synthetic)	Blackbox Exporter	node_exporter	kube-state-metrics
Datadog (awareness)	New Relic (awareness)	Splunk (awareness)	Ollama (Bonus AI)

09 What Makes This Different

OTHER COURSES	THIS COURSE
Teach a single vendor (Datadog, Splunk)	Vendor-neutral — defend OSS or commercial
Three pillars slides	Working OpenTelemetry + Prometheus + Loki + Tempo stack you build yourself
Toy examples (hello world service)	A realistic banking application with real failure modes
Show you the UI	Make you write PromQL under simulated incident pressure
Stop at metrics + logs	Continuous profiling, eBPF, incident response — full senior skill set
No artifacts	Portfolio on GitHub you link from your resume

10 What's Included

- **33 production-grade module guides** — concepts, war stories, labs, pitfalls, cheat sheets
- **Manifest library** — ServiceMonitors, PrometheusRules, Helm values
- **Runbook templates** + a written postmortem template
- **Cheat sheets** — PromQL, LogQL, TraceQL, OTel env vars, SLO math
- **Complete bankobserve360 lab app** with chaos-injection hooks
- **Exportable Grafana dashboards** (JSON) for every lab
- **Interview question bank** with model answers, by topic
- **Tools matrix** — when to pick what, with reference architectures

11 Format & Prerequisites

Format & Delivery

- 4 weeks, ~25 hours (~6 hrs/week)
- Live sessions with async support
- Cohort size capped for quality
- Private GitHub repo (markdown + manifests)
- Lifetime access to materials + lab app
- Course Slack community included

Prerequisites

- Comfortable on Linux CLI
- Basic Docker and Kubernetes (kubectl)
- Git
- Laptop with 16 GB RAM (8 GB min)
- Docker Desktop + ~50 GB free disk
- No cloud account needed (runs on kind)

12 Bonus Modules

Included at no extra cost with your enrollment:

BONUS 01

Resume + LinkedIn Rewrite Guide

How to position your new portfolio for senior roles.

BONUS 02

Mock Interview Question Bank

40+ questions across 11 topic areas, with model answers.

BONUS 03

bankobserve360 Chaos Scenario Library

Additional failure modes for ongoing practice after the course ends.

BONUS 04

Course Slack Community

Alumni network and ongoing Q&A, included in your enrollment.

BONUS 05

Custom Collector for Certificates & Licenses

Build a Prometheus-compatible collector that scrapes TLS certificate expiry and software license utilization across your fleet — wired into Grafana with burn-rate-style alerts.

BONUS 06

Business-Level Metrics & Operations Dashboards

Instrument UPI volume, loan disbursement, NPA trending, KYC funnel, and fee income — build the Bank Operations dashboard that gets engineers into exec-staff meetings.

BONUS 07

AI-Assisted Observability Workflows (Without Getting Burned)

Natural language to PromQL/LogQL/TraceQL generation, postmortem drafting from incident transcripts, and a critical-literacy drill on AI failure modes.

13 About the Instructor

RK

Raghu K

Built and operated observability platforms for **17 years** across enterprise environments. Has trained **9,000+ engineers** to date.

14 Frequently Asked Questions

Q How is this different from the SRE book?

The SRE book is the conceptual foundation. This course is the applied version — the book teaches what an SLO is, this course makes you implement one with multi-burn-rate alerts.

Q Do I need to know Prometheus already?

No. Module 1.2 starts from 'what is a metric.' If you've used Prometheus, you'll move faster; if not, you'll be fluent by end of Week 1.

Q What if I use Datadog at work, not Prometheus?

Even better. Module 4.5 covers Datadog/New Relic/Dynatrace explicitly. The skills transfer to every backend — OpenTelemetry is the bridge.

Q Will this help me pass an SRE interview?

That's a primary design goal — interview-grade questions in every module, a dedicated interview bank, and a system-design rehearsal in Module 4.5.

Q How much hands-on work is there?

Every module has hands-on work. The capstone is a 90-minute simulated incident. Labs alone are ~12 of the 25 hours.

Q Do I need a cloud account?

No. Everything runs locally on kind (Kubernetes in Docker) on your laptop.

Q What happens after the course?

You leave with a GitHub portfolio, a 90-day reading list, and an interview bank you can drill on. The skills compound.

joindevops.com

• READY WHEN YOU ARE

Stop being the person who uses tools.
Become the person who picks them.

PROGRAM FEE

₹8,990/-

EMI options available through Credit Cards

EMAIL	CALL / WHATSAPP	WEBSITE
info@joindevops.com	6281937079	joindevops.com

@DevOpsAndCloudWithSiva · @JoinDevOps-siva

Batch details, start dates, and live session timings are shared upon enrollment inquiry.
Reach out to reserve your seat for the next cohort.