OpenTelemetry – Complete Deep-Dive Guide

1. What is OpenTelemetry

OpenTelemetry (OTel) is a vendor-neutral, open-source observability framework designed to standardize how applications generate, collect, and export telemetry data.

2. Why OpenTelemetry Exists

OpenTelemetry solves the problem of fragmented observability tooling by unifying telemetry standards and eliminating vendor lock-in.

3. High-Level Architecture

```
Application
   |
OTel API
   |
OTel SDK
   |
Exporter
   |
OTel Collector
   |
Backend (Prometheus / Jaeger / Grafana)
```

4. Telemetry Signals

```
Traces  -> End-to-end request flow
Metrics -> Numerical system health data
Logs    -> Structured event details
```

5. Traces – Deep Dive

```
Trace
 |
 +-- Span: API
 |
 +-- Span: Auth Service
 |
 +-- Span: Order Service
 |
 +-- Span: Payment Service
```

6. Metrics – Deep Dive

Metrics include counters, gauges, and histograms. They provide trend analysis and support alerting.

7. Logs – Deep Dive

Logs become more powerful when correlated with tracing context such as Trace ID and Span ID.

8. Context Propagation

```
traceparent: version-traceid-spanid-flags
```

9. OTel Collector Architecture

```
Receivers -> Processors -> Exporters
```

10. Data Flow Lifecycle

```
Request -> Span -> Context Propagation -> SDK Batching -> Collector Processing -> Backend Storage
```

11. Deployment Models

- Agent Mode - Gateway Mode - Hybrid Mode

12. Best Practices

Use OTel Collector in production, avoid high-cardinality metrics, enable sampling, and combine auto+manual instrumentation.

13. Common Mistakes

Not using the collector, missing context propagation, and collecting excessive data without sampling.