OpenTelemetry Training and Certification Course

Curriculum 5 Days

Day - 1

- **Welcome and Overview of the Course**
  - Course objectives and expectations.
  - Brief introduction to observability (tracing, metrics, logging).
  - Why OpenTelemetry? Benefits of using OpenTelemetry in modern cloud-native applications.
- **Fundamentals of Observability**
  - Course objectives and expectations.
  - Brief introduction to observability (tracing, metrics, logging).
  - Why OpenTelemetry? Benefits of using OpenTelemetry in modern cloud-native applications.
- **Introduction to OpenTelemetry**
  - What is OpenTelemetry?
  - OpenTelemetry's components (SDK, API, Exporters, Collector).
  - Supported languages and frameworks.
  - The OpenTelemetry ecosystem: Integration with tracing systems (e.g., Jaeger, Zipkin), metrics (Prometheus), and logging (Fluentd, ELK).
- **Installing OpenTelemetry SDK**
  - Setting up the OpenTelemetry SDK in a programming language of choice (e.g., Java, Python, Go, JavaScript).
  - Overview of installation steps and basic configurations.
  - Hands-on Lab: Installing and configuring the OpenTelemetry SDK.

Day - 2

- **Introduction to Distributed Tracing**
  - Traces vs. Metrics vs. Logs: Why Tracing is crucial for observability.
  - Key tracing concepts: Spans, Context propagation, Trace IDs, and Parent-child relationships.
  - How tracing works in a distributed system (from HTTP requests to database calls).
- **Manual vs. Automatic Instrumentation**
  - Using OpenTelemetry’s automatic instrumentation for common libraries.
  - Manually instrumenting code (e.g., HTTP servers, database clients).
  - Example: Manual instrumentation in Python/Java/Go.
- **Instrumenting Applications with OpenTelemetry**
  - Hands-on Lab: Instrumenting a microservice application.
  - Trace generation, context propagation, and adding custom spans.
  - Integrating OpenTelemetry with popular frameworks (e.g., Flask, Spring Boot).
- **Exporting Traces to Backend Systems**
  - Setting up trace exporters to send data to Jaeger, Zipkin, or OpenTelemetry Collector.
  - Using OpenTelemetry Collector to aggregate and forward traces.
  - Hands-on Lab: Setting up trace exporters and viewing trace data in Jaeger.

Day - 3	Day - 4

• **Introduction to Metrics**
  ○ What are metrics? Different types (counters, histograms, gauges).
  ○ Understanding Prometheus and how OpenTelemetry integrates with it.
  ○ Collecting and exposing application metrics via OpenTelemetry.
• **Instrumenting for Metrics**
  ○ Hands-on Lab: Instrumenting a simple application for basic metrics collection.
  ○ Collecting latency, request count, error rates, and other key metrics.
• **Exporting Metrics**
  ○ Setting up Prometheus exporter for OpenTelemetry metrics.
  ○ Viewing metrics in Grafana or Prometheus.
  ○ Exploring metric data and setting up alerts based on thresholds.
• **Advanced Metrics: Custom Metrics and Aggregation**
  ○ Creating custom metrics.
  ○ Working with metrics aggregation and filtering in OpenTelemetry Collector.
  ○ Hands-on Lab: Creating custom metrics and visualizing them in Grafana.
	• **Deep Dive into OpenTelemetry Collector**
	  ○ Introduction to OpenTelemetry Collector and its architecture.
	  ○ Setting up an OpenTelemetry Collector to collect, process, and export telemetry data.
	  ○ Configuring the Collector for advanced use cases (e.g., aggregation, batching, filtering).
	• **Scaling Observability with OpenTelemetry**
	  ○ Using OpenTelemetry Collector for large-scale applications.
	  ○ Best practices for scaling and managing OpenTelemetry in production environments.
	  ○ Hands-on Lab: Setting up and configuring OpenTelemetry Collector with multiple exporters.
	• **OpenTelemetry in Kubernetes and Cloud-Native Environments**
	  ○ Instrumenting applications running in Kubernetes.
	  ○ Using OpenTelemetry with containerized workloads.
	  ○ Integrating with cloud-native monitoring tools (e.g., Prometheus, Grafana, and cloud-based backends like AWS X-Ray, GCP Trace).
	• **Using OpenTelemetry with Other Observability Tools**
	  ○ Integrating OpenTelemetry with existing monitoring setups (e.g., ELK stack, Datadog).
	  ○ Hands-on Lab: Configuring OpenTelemetry with Kubernetes and cloud-native observability tools.

Day - 5

- **Debugging and Troubleshooting with OpenTelemetry**
  - Common issues when working with OpenTelemetry (instrumentation errors, missing traces, metric inconsistencies).
  - Techniques for debugging OpenTelemetry instrumentation and tracing data.
  - Using trace data for troubleshooting performance bottlenecks and errors.
- **Best Practices for Using OpenTelemetry in Production**
  - Best practices for instrumenting applications at scale.
  - Understanding performance implications of telemetry collection.
  - Efficient data collection and storage practices.
- **Security and Privacy Considerations in Telemetry Data**
  - Ensuring security when exporting telemetry data (e.g., encryption).
  - Privacy concerns with telemetry data and how to address them.
- **OpenTelemetry Roadmap and Community**
  - What's next for OpenTelemetry? Upcoming features and developments.
  - How to contribute to the OpenTelemetry project.
  - OpenTelemetry community and resources.
- **Hands-on Lab: Final Project**
  - Building a fully instrumented and monitored application using OpenTelemetry (traces + metrics).
  - Exporting the data to a backend system (e.g., Jaeger for tracing, Prometheus for metrics).
  - Group discussion and troubleshooting of final projects.