Faysal Aziz

# Deployment for DevOps

## 2-day intensive bespoke advanced course

FA-0164 · 2-day

DevOps, Cloud & Infrastructure

---

## Introduction

In the rapidly evolving world of software development and operations (DevOps), observability has become a cornerstone of ensuring application reliability, performance, and security. Observability allows teams to gain deep insights into their systems through the collection, analysis, and visualization of logs, metrics, and traces. It empowers developers and operators to identify and resolve issues swiftly, improving overall system robustness and user satisfaction. Complementing observability, Helm charts have revolutionized the deployment and management of Kubernetes applications, providing a standardized and efficient way to package, configure, and deploy complex applications in containerized environments.

This course aims to provide a comprehensive understanding of observability and Helm charts, equipping participants with the knowledge and skills needed to enhance their development and DevOps practices. Participants will learn log analysis, metrics visualization, tracing data analysis, and how to leverage Helm charts for effective application deployment. By the end of the course, attendees will be well-versed in modern observability techniques and proficient in using Helm charts to streamline Kubernetes deployments.

## Learning outcomes

- Understand the importance of observability in development and DevOps.
- Master log analysis and search techniques to troubleshoot issues.
- Gain proficiency in metrics analysis and visualization for performance monitoring.
- Learn to analyze tracing data for identifying and troubleshooting bottlenecks.
- Develop skills in correlation and root cause analysis for effective problem resolution.
- Explore anomaly detection and predictive analysis to preempt potential issues.
- Understand Helm charts and their role in Kubernetes application management.
- Determine when and why to use Helm and Helm charts.
- Gain practical experience in getting started with Helm charts.

# Prerequisites

- Basic understanding of Kubernetes and containerization concepts.
- Familiarity with DevOps practices and principles.
- Basic knowledge of logging, metrics, and tracing in software applications.
- Experience with the command line and basic scripting.

# Training outline

## 1. Observability for Development and DevOps

**Introduction to Observability**

- Definition and significance in modern software development
- Key components: logs, metrics, and traces
- Differences between monitoring and observability

**Log Analysis and Search Techniques**

- Importance of log analysis in troubleshooting
- Common logging frameworks and tools (e.g., ELK stack, Fluentd, Loki)
- Log collection and aggregation
- Search techniques and query languages (e.g., Lucene query language, Fluent query language)
- Practical exercises on searching and filtering logs

**Metrics Analysis and Visualization**

- Understanding metrics and their importance in performance monitoring
- Common metrics types (e.g., counters, gauges, histograms)
- Tools for metrics collection and visualization (e.g., Prometheus, Grafana)
- Setting up and configuring Prometheus for metrics collection
- Creating and interpreting Grafana dashboards
- Practical exercises on metrics visualization

**Tracing Data Analysis and Troubleshooting**

- Introduction to distributed tracing
- Benefits of tracing in microservices architecture
- Common tracing tools (e.g., Jaeger, Zipkin, OpenTelemetry)
- Setting up tracing in a sample application
- Analyzing trace data to identify bottlenecks and performance issues
- Practical exercises on tracing data analysis

**Correlation and Root Cause Analysis**

- Importance of correlation in observability
- Techniques for correlating logs, metrics, and traces
- Tools for correlation and root cause analysis
- Practical exercises on root cause analysis

**Anomaly Detection and Predictive Analysis**

- Understanding anomalies and their impact on system performance
- Techniques for anomaly detection (e.g., statistical methods, machine learning)
- Tools for anomaly detection and predictive analysis
- Implementing anomaly detection in a sample application
- Practical exercises on predictive analysis

## 2. Helm Charts

**Explore Helm Charts**

- Introduction to Helm and Helm charts
- Structure and components of a Helm chart
- Helm repositories and how to use them
- Practical exercises on exploring Helm charts

**Determine When and Why to Use Helm and Helm Charts**

- Benefits of using Helm charts for Kubernetes applications
- Use cases and scenarios where Helm charts are advantageous
- Comparison with other deployment methods
- Practical discussions and case studies

**Provide Instructions for Getting Started with Helm**

- Installing Helm and setting up a Helm environment
- Creating and managing Helm charts
- Deploying applications using Helm charts
- Upgrading and rolling back Helm releases
- Practical exercises on getting started with Helm

This detailed and comprehensive outline ensures that participants gain a thorough understanding of both observability and Helm charts, with ample practical exercises to reinforce the concepts learned. By the end of the course, attendees will be equipped with the skills needed to implement effective observability practices and utilize Helm charts for efficient Kubernetes application management.

© 2026 Faysal Aziz · Kuala Lumpur