Advanced Platform Engineering: Scaling with Microservices and Kubernetes

Duration – 5 Day

Overview

This is an intensive training program designed for advanced-level platform engineers and DevOps professionals. The "Advanced Platform Engineering" course focuses on the strategic application of microservices and Kubernetes to build and scale applications efficiently and reliably.

This instructor-led, live training is a hands-on, immersive experience. By the end of this training, the participant will have a deep, practical understanding of designing, deploying, and managing complex, scalable systems in a modern cloud-native environment.

Course Objectives

By the end of this training, participants will be able to:

- Design and implement scalable microservices architectures.
- Deploy and manage complex applications on production-grade Kubernetes clusters.
- Utilize Helm charts for templating and managing service deployments efficiently.
- Implement robust observability stacks to monitor and maintain the health of microservices in production.
- Apply industry best practices for security and compliance within a Kubernetes environment.
- Architect resilient systems that can handle failure gracefully..

Target Audience

Audience: Platform Engineers, DevOps Professionals, and Software Architects.

Prerequisites:

- A solid understanding of containerization and orchestration concepts.
- Practical experience with Docker and basic Kubernetes operations.
- Familiarity with cloud computing principles and microservices architecture.

Training Outline

Day 1: Foundations of Scalable Architectures

- Module 1: Advanced Concepts in Platform Engineering
    - Review of Platform Engineering fundamentals.
    - The evolution of platform engineering in the context of scalability.
    - Design principles for a scalable and resilient platform architecture.
- Module 2: Deep Dive into Microservices Architecture
    - Core principles and patterns of microservices design.

- Practical lab: Decomposing a monolithic application into microservices.
- Communication patterns (Synchronous vs. Asynchronous, REST, gRPC, Message Queues).
- Service discovery mechanisms and implementation.

Day 2: Containerization and Introduction to Kubernetes

- Module 3: Advanced Containerization with Docker
  - Advanced containerization techniques and best practices.
  - Lab: Building optimized, multi-stage Docker images for microservices.
  - Deep dive into container networking and storage considerations.
- Module 4: The Kubernetes Ecosystem and Architecture
  - In-depth exploration of Kubernetes components (Control Plane, Worker Nodes, etcd).
  - Lab: Setting up and managing a Kubernetes cluster from scratch.
  - Kubernetes networking (CNI), storage (CSI, Persistent Volumes), and security best practices.

Day 3: Deploying and Managing Applications on Kubernetes

- Module 5: Deploying Microservices on Kubernetes
  - Understanding Kubernetes workload resources (Deployments, StatefulSets, DaemonSets).
  - Advanced deployment strategies: Blue/Green, Canary, and Rolling updates.
  - Lab: Implementing various deployment strategies for a sample microservice.
- Module 6: Managing Services with Helm
  - Introduction to Helm charts for packaging and deployment.
  - Lab: Creating a Helm chart from scratch for a multi-service application.
  - Continuous deployment (CD) practices and integrating Helm into a CI/CD pipeline.

Day 4: Observability, Scaling, and Performance

- Module 7: Observability and Monitoring in Microservices
  - The three pillars of observability: Logging, Metrics, and Tracing.
  - Implementing a monitoring stack (e.g., Prometheus, Grafana).
  - Lab: Setting up logging, monitoring, and alerting for a microservices application.
  - Analyzing metrics and logs to maintain system health and troubleshoot issues.
- Module 8: Scaling and Performance Tuning
  - Horizontal (HPA) and Vertical (VPA) Pod Autoscaling.
  - Cluster auto-scaling strategies.
  - Lab: Configuring auto-scaling for services based on CPU and memory load.
  - Performance tuning and resource optimization (requests, limits, quality of service).

Day 5: Resilience, Security, and Final Project

- Module 9: Resilience and Reliability Engineering
  - Designing for failure: Resilience patterns (Circuit Breakers, Retries, Timeouts).
  - Implementing health checks, readiness probes, and liveness probes.
  - Disaster recovery and high-availability strategies for Kubernetes clusters.
- Module 10: Security for Microservices and Kubernetes
  - Securing microservice communications (mTLS).

- Kubernetes security features (RBAC, Pod Security Policies, Network Policies).
- Introduction to service meshes (e.g., Istio, Linkerd) for enhanced security and observability.

- Module 11: Hands-On Capstone Project
  - **Objective:** Design, deploy, and manage a scalable, secure, and observable microservices application on Kubernetes.
  - The participant will apply concepts from the entire week to:
    1. Deploy a multi-service application using a CI/CD pipeline.
    2. Configure monitoring and alerting.
    3. Implement scaling and security policies.
    4. Demonstrate the application's resilience by simulating failures.