CL400 Cloud-native Software Development

Kurzbeschreibung:

Der Workshop **CL400 Cloud-native Software Development** fasst in einzigartiger Weise zusammen, was Sie als Architekt, Developer oder DevOps-Engineer über diese wegweisende Methode zur Entwicklung skalierbarer und sicherer Softwarelösungen wissen müssen.

Cloud-native Anwendungen sind aktuell State-of-the-Art, um Anwendung in der Private Cloud oder Public Cloud zu entwickeln. Sie sind skalierbar, haben erhöhte Sicherheit und ihre Elastizität ist auf die dynamische Nutzung von Cloud-Ressourcen ausgerichtet.

Cloud-native Anwendungen werden in der Regel in agilen Kontexten entwickelt, in denen Konzepte wie DevOps, GitOps oder Platform Engineering gelebt werden. Die Anwendungen werden oft auf Basis von Frameworks und Programmiersprachen wie Spring Boot, Go, Python oder Node.js entwickelt und als Container bereitgestellt.

Um Sie in der Cloud zu deployen, wird Infrastruktur benötigt. Hier bietet Terraform als Infrastructure-as-Code Lösung einen geeigneten Ansatz, diesen zu provisionieren. Viele Anwendungen laufen auf Kubernetes, das durch DevOps-Teams auch betrieben werden können sollte. Um Anwendungen auf Kubernetes zu deployen, sollten sie paketiert werden. Das Framework Helm vbietet hier eine Lösung.

Damit ein automatisiertes Bauen und Deployen der Anwendung möglich ist, bedarf es einer CI/CD Strecke. Dies ist mit den Bausteinen Tekton und Argo CD möglich. Prometheus und Grafana helfen im Betrieb beim Monitoring.

Zielgruppe:

Das Seminar **CL400 Cloud-native Software Development** richtet sich an:

- Architekten
- Entwickler
- DevOps-Engineers

Voraussetzungen:

Um dem Lerntempo und Inhalten des Trainings **CL400 Cloud-native Software Development** gut folgen zu können, sind folgende Voraussetzungen zu erfüllen:

- Linux Systemadministration Basics
- Erfahrungen in einer Programmiersprache
- Grundlegende Erfahrung in Netzwerkadministration
- Grundlagen in Cloud Computing

Sonstiges:

Dauer: 4 Tage

Preis: 3250 Euro plus Mwst.

Ziele:

- Erlangung von Wissen über Cloud-native Anwendungen
- Architektur-Verständnis von Cloud-native Anwendungen
- Implementierung von Cloud-native Anwendungen auf Basis von populären Frameworks und Tools

Inhalte/Agenda:

- ♦ **Überblick und Begriffsklärung**
  - ♦ ◊ DevOps
    ◊ Platform Engineering
    ◊ SRE
    ◊ GitOps
- ♦ ◊
- ♦ **Einführung in Micro-Service Programmierung**
  - ♦ ◊ Java Spring Boot
    ◊ Go
    ◊ Python
    ◊ Nodejs
- ♦ ◊
- ♦ **Containerisierung von Microservices**
  - ♦ ◊ Dockerfiles
    ◊ Build-Packs
    ◊ Maven-Jib Plugins
- ♦ ◊
- ♦ **Einführung in Infrastructure-as-Code (IaC) mit Terraform**
  - ♦ ◊ Motivation IaC
    ◊ Abgrenzung
    ◊ Terraform Getting Startet
    ◊ Terraform Provider
    ◊ Terraform Variablen
    ◊ Terraform Module
    ◊ Überblick über kommerzielle Terraform Produkte
- ♦ ◊
- ♦ **Einführung in Kubernetes**
  - ♦ ◊ Architektur von Kubernetes
    ◊ Deployments
    ◊ Services
    ◊ Ingress Controller und Ingress
    ◊ Logging von Kubernetes
    ◊ Skalierung
    ◊ Kubernetes Cluster Verwaltung
- ♦ ◊
- ♦ **Paketierung von Kubernetes Anwendungen**
  - ♦ ◊ Einführung in Helm Charts
    ◊ Kustomize als Overlay Technologie
- ♦ ◊
- ♦ **Cloud-native CI am Beispiel von Tekton**
  - ♦ ◊ Einführung und Architektur Tekton
    ◊ Getting Started with Tekton
    ◊ Tekton Pipelines und Trigger
    ◊ Supply Chain Security mit Tekton
    ◊ Bauen von Images mit Tekton
- ♦ ◊
- ♦ **Einführung von GitOps am Beispiel von ArgoCD**
  - ♦ ◊ Motivation GitOps
    ◊ Einführung und Architektur ArgoCD
    ◊ GitOps mit Kubernetes
    ◊ Deployment von Microservices mit GitOps
    ◊ Secret Management mit ArgoCD
- ♦ ◊
- ♦ **Observability**
  - ♦ ◊ Motivation und Einführung in Logging und Monitoring
    ◊ Monitoring mit Prometheus und Grafana
    ◊ Übersicht über Logging von Cloud-Native Lösungen
  - ♦ ◊