

CL400 Cloud-native Software Development

Kurzbeschreibung:

Architekten, Developer und DevOps-Engineers erhalten eine praxisnahe Einführung in Cloud-native Softwareentwicklung. Behandelt werden Frameworks wie Spring Boot, Go, Python oder Node.js sowie Containerisierung und Deployment auf Kubernetes mit Helm. Ergänzt wird das Training durch Infrastructure-as-Code mit Terraform, CI/CD-Pipelines mit Tekton und Argo CD sowie Monitoring mit Prometheus und Grafana.

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:

Nach Abschluss des Kurses **CL400 Cloud-native Software Development** verfügen Sie über folgende Kenntnisse und Fähigkeiten:

- Wissen über Cloud-native Anwendungen
- Architektur-Verständnis von Cloud-native Anwendungen
- Fähigkeiten zur 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
 - ◆ ◊