

## **CL400-WS Cloud-native Development - Platform Engineering in der Praxis**

### **Kurzbeschreibung:**

Teilnehmende erhalten eine praxisnahe Einführung in Cloud-native Development mit Fokus auf Kubernetes, Microservices und DevOps-Konzepte. Behandelt wird, wie mittels GitOps-Technologien eine cloud-native Anwendung in einem Kubernetes-Cluster bereitgestellt werden kann. Das Seminar zeigt, wie Skalierbarkeit, Ausfallsicherheit und moderne Prozessstrategien in der Praxis umgesetzt werden.

### **Zielgruppe:**

Das Web-Seminar **CL400-WS Cloud-native Software Development - Platform Engineering in der Praxis** richtet sich an:

- Architekten
- Entwickler
- DevOps-Engineers

### **Voraussetzungen:**

Cloud-native development is a modern approach to building software solutions. This method is specifically tailored to cloud infrastructures and leverages the flexibility, scalability, and resilience of the cloud. Key technologies and architectural models include Kubernetes and microservices. In addition, agile methods and DevOps concepts are applied as part of a process strategy.

In the web seminar CL400-WS Cloud-native Development – Platform Engineering in Practice, we demonstrate how a cloud-native application can be deployed in a Kubernetes cluster using GitOps technologies.

### **Sonstiges:**

**Dauer:** 1 Tage

**Preis:** 0 Euro plus Mwst.

### **Ziele:**

Im Web-Seminar **CL400-WS Cloud-native Development - Platform Engineering in der Praxis** erfahren Sie, wie eine cloud-native Anwendung in einem Kubernetes Cluster mittels GitOps-Technologien deployt werden kann.

Der Kurs **CL400 Cloud-native Software Development** vermittelt das Wissen, ein derartiges Deployment selbst zu bauen.

## Inhalte/Agenda:

- ♦ Überblick und Begriffsklärung
- ♦ Einführung in Micro-Service Programmierung
- ♦ Containerisierung von Microservices
- ♦ Einführung in Infrastructure-as-Code (IaC) mit Terraform
- ♦ Einführung in Kubernetes
- ♦ Paketierung von Kubernetes Anwendungen
- ♦ Cloud-native CI am Beispiel von Tekton
- ♦ Einführung von GitOps am Beispiel von ArgoCD
- ♦ Observability