

## **LI530 Kubernetes Advanced**

### **Kurzbeschreibung:**

Der Workshop **LI530 Kubernetes Advanced** richtet sich an erfahrene Linux/Unix-Administratoren mit soliden Kenntnissen in Kubernetes und Linux. Er vermittelt tiefgehende Einblicke in fortgeschrittene Kubernetes-Themen wie die Bereitstellung von Applikationen in Containern, Deployment- und Template-Engines mit Helm, sowie den Aufbau und die Verwaltung von Helm-Charts. Ein Schwerpunkt liegt auf Sicherheit: Teilnehmer lernen Zugriffskontrollen auf die Kubernetes API, verschiedene Authentifizierungsverfahren und Cluster-Sicherheitsrichtlinien wie PodSecurityPolicy und NetworkPolicies kennen und anwenden.

Ergänzt wird das Seminar durch praxisorientierte Übungen zu Image-Sicherheit, CVE-Scanning und Container-Signierung. Zudem erhalten die Teilnehmer einen umfassenden Überblick über ServiceMesh-Technologien mit Schwerpunkt Istio und die Funktionsweise sowie Erstellung von Kubernetes Operatoren. Nach Abschluss sind sie befähigt, diese Konzepte sicher und effizient in Projekten einzusetzen.

Wir empfehlen als Aufbaukurse:

- [LI560 Kubernetes Security](#)
- [LI590 Kubernetes Cluster Management mit SUSE Rancher](#)

### **Zielgruppe:**

Das Seminar **LI530 Kubernetes Advanced** richtet sich in erster Linie an Linux/Unix Administratoren.

### **Voraussetzungen:**

Um dem Kursinhalten und dem Lerntempo des Workshops **LI530 Kubernetes Advanced** gut folgen zu können, sind stabile Vorkenntnisse in den Bereichen Linux und Kubernetes nötig.

Wir empfehlen Ihnen den vorherigen Besuch der folgenden Kurse:

- [LI500 Container Grundlagen](#)
- [LI510 Kubernetes Basics](#)

### **Sonstiges:**

**Dauer:** 5 Tage

**Preis:** 2990 Euro plus Mwst.

### **Ziele:**

Nach Abschluss des Kurses **LI530 Kubernetes Advanced** sind die Teilnehmer in der Lage, die vorgestellten Konzepte im Besonderen das Sicherheitskonzept und darauf aufbauend die ServiceMesh Technologie in eigenen Projekten sinnvoll einzusetzen.

## Inhalte/Agenda:

- **◆ Bereitstellung von Applikationen im Container**
  - ◆ Übersicht der Deployment und Template Engines
  - ◆ Helm, Charts und Repo
  - ◆ Aufbau von Helm Charts
  
- **◆ Zugriffsberechtigungen auf die Kubernetes API**
  - ◆ Übersicht:
    - ◆ · Service Account Tokens (JWT)
    - ◆ · OpenID Connect Tokens (JWT)
    - ◆ · Webhook Token Authentication
  - ◆ Beispiele und Übungen
  
- **◆ Sicherheitseinstellungen für den Kubernetes-Cluster**
  - ◆ Absicherung der Kubernetes-Serverkomponenten
  - ◆ Policies im Kubernetes-Cluster
    - ◆ · PodSecurityPolicy
    - ◆ · NetworkPolicies
    - ◆ · Webhook Admission Controller
    - ◆ · Beispiele und Übungen
  - ◆ Image & Container Security
    - ◆ · Container Image CVE Scanning
    - ◆ · Container Image signieren
  
- **◆ ServiceMesh**
  - ◆ Was ist ein ServiceMesh?
  - ◆ Übersicht
  - ◆ ServiceMesh am Beispiel Istio
  
- **◆ Kubernetes Operatoren**
  - ◆ Funktionsweise von Operatoren in Kubernetes
  - ◆ Erstellen eines Operators