*Trainings*

# Kubernetes Security Fundamentals


This Kubernetes security training course examines the security aspects of the entire cluster lifecycle: from initial configuration and deployment through operations to dealing with newly discovered or published vulnerabilities (CVEs).

Dauer: 4 days

Zielgruppe: Kubernetes administrators with good Linux knowledge

Kubernetes has established itself in recent years as the standard framework for the infrastructure underlying SaaS (Software as a Service) products. Almost all companies make Kubernetes clusters available to their developers as operating environments, whether in public or private clouds. One essential prerequisite for the *secure* operation of such SaaS products is a *secure* Kubernetes cluster. A Kubernetes cluster is, however, a complex system in which merely installing the latest security patches is not enough.

This training course examines the security aspects of the entire cluster lifecycle: from initial configuration and deployment through operations to dealing with newly discovered or published vulnerabilities (CVEs). The course will also discuss how deployment pipelines for SaaS products can be designed to prevent typical supply chain attacks.

Like all training courses offered by the inovex Academy, this course focuses on practical learning. During the course, participants will design and build their own *secure* Kubernetes cluster on which they will implement security monitoring and logging.

## Target Audience:

Kubernetes administrators with good Linux knowledge. Prerequisite is either the Linux Foundation's CKA certification, a certificate of participation in the LFS458 training course, or equivalent knowledge of the structure and function of Kubernetes clusters.

# Agenda:

- Introduction to cloud security concepts
  - Assessment
  - Prevention
  - Detection
  - Reaction
  - Attack surfaces and types
- Secure cluster configuration
  - Image supply chains
  - Policy-based control
  - Runtime sandbox
- Cluster installation
  - Updating Kubernetes
  - Kernel hardening
- Securing kube-apiserver
  - Access restrictions
  - Role-based access control
  - Auditing
  - Protecting etcd
- Secure Networking
  - Netfilter management and implementation

- Pod-to-pod encryption
- Restricting cluster-level access

- Security and workloads
  - Analysing workloads
  - SELinux basics
  - Implementing AppArmor

- Detecting and handling security incidents
  - Implementing intrusion detection systems
  - Best practices during an incident
  - Best practices following an incident
  - Threat detection
  - Behaviour analysis