

## **KBS-222: Kubernetes Advanced Administration (Helm & Security)**

**Course Length:** 1 day Helm + 1 day Cloud-native security, 2 days altogether

### **Course Description:**

Kubernetes is the de-facto system for container orchestration, e.g. automating the deployment, scaling and management of microservices-based, containerized applications.

This course builds on the of Linux container and basic Kubernetes admin. skills participants gained on our Docker and Kubernetes trainings or in practice and enhances their knowledge with more advanced Kubernetes features and extensions such as the Helm package manager as well as major cloud-native security topics and tools to handle security related tasks.

**Structure:** 50% theory 50% hands on lab exercises

**Target audience:** System administrators, developers and devops with Kubernetes admin. skills who want to learn about advanced Kubernetes extensions like the Helm package manager as well as major cloud-native security topics and tools to handle security related tasks.

**Prerequisites:** Linux container (e.g. Docker) and Kubernetes admin. skills, for instance by participating on our Docker and Kubernetes administration courses.

### **Detailed Course Outline**

#### **PART I: Helm Package Manager**

##### **Module 1: Introduction to Helm**

- What is helm?
- Main Helm Concepts
- Helm Components
- Helm Implementation

##### **Module 2: Installing and securing Helm and Tiller**

- Installing Helm
- Installing Tiller
- Securing Tiller
- Best practices for securing Tiller and Helm

##### **Module 3: Using Helm**

- Generic options
- Working with repositories
- Finding charts
- Installing a release
- List releases
- Upgrade/rollback releases
- Deleting releases

## **Module 4: Helm Charts**

- Introduction to charts
- The structure
- The Chart.yaml File
- The components of a Chart
- Chart dependencies
- Chart dependencies (cont.)
- Chart lifecycle hooks
- Managing charts with Helm

## **Module 5: Chart Templates**

- Writing Templates
- Templates and Values
- Dependencies and values
- Dependencies and values
- Functions and pipelines
- Flow control
- Variables
- Named templates

## **Module 6: Helm plugins**

- Building Plugins

## **PART II: Cloud-native Security**

### **Module 7: Managing users in Kubernetes (Authentication and permissions)**

- Establishing the identity
- Using OIDC tokens with Kubernetes
- Creating our own OIDC provider with Dex
- Authorization in Kubernetes
- Role based access control (RBAC)
- Lab: Authenticate LDAP users into Kubernetes

### **Module 8: Secure image registries**

- Security requirements for image registries
- Building secure registries using Harbor
- Harbor Operations
- Lab: Using Harbor as an image registry

## **Module 9: Penetration testing the Kubernetes API**

- What to test?
- Tools for penetration testing
- Lab: Penetration test the Kubernetes cluster

## **Module 10: Monitoring application behaviour**

- Mechanisms for controlling application behavior
- Tools for observing application behavior
- Observing application behavior using Falco
- Lab: Monitor application behavior in Kubernetes

## **Module 11: Container isolation mechanisms**

- Pod security policies
- Configuring pod security policies
- Lab: Enforce security rules using Pod Security Policies