ComponentSoft
open cloud trainings and consultation

KBS-535: Kubernetes administration & Kubernetes and Container-based Application Security with CKS exam.prep.

Course Length: 3 days Kubernetes + 2 day Cloud-native security, 5 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 training first introduces participants to the basic concepts and architecture of Kubernetes, its initial install, setup and access control, Kubernetes Pods and Workloads, Scheduling and node management, Accessing the applications, Persistent storage in Kubernetes as well as its Logging, Monitoring and Troubleshooting facilities.

The second part enhances the delegates' knowledge with information about the most important Kubernetes and container related security topics and tools. It introduces concepts, procedures, and best practices to harden Kubernetes based systems and container-based applications against security threats. It deals with the main areas of cloud-native security: Kubernetes cluster setup, Kubernetes cluster hardening, hardening the underlying operating system and networks, minimizing microservices vulnerabilities, obtaining supply chain security as well as monitoring, logging, and runtime security.

This course doesn't only deal with the daily and security administration of Kubernetes based systems but also prepares delegates for the official Certified Kubernetes Security Specialist (CKS) exam of the Cloud Native Computing Foundation (CNCF).

Structure: 50% theory 50% hands on lab exercises

Target audience: System administrators, developers and devops who want to understand and use Kubernetes in enterprise and cloud environments.

Prerequisites: Proficiency with the Linux CLI. A broad understanding of Linux system administration. Basic knowledge of Linux containers, e.g. Docker.

Detailed Course Outline

PART I. Kubernetes Administration

Module 1: Kubernetes introduction

- Cloud computing in general
- Cloud types
- Cloud native computing
- Container orchestration
- Kubernetes
- Kubernetes concepts
- Kubernetes objects categories
- Custom resource definitions
- Kubernetes architecture
- Kubernetes master
- Kubernetes node
- Kubernetes Lab: Health check

Module 2: Accessing Kubernetes

- Accessing the Kubernetes cluster
- Controlling access to the API
- Authorization
- Role Based Access Control
- Roles and ClusterRoles
- Role bindings
- Admission control
- Kubernetes Lab: Accessing API

Module 3: Kubernetes Workloads

- The pod
- Our first Pod
- Operations on pods
- Pod Status and Lifecycle Pod Status and Lifecycle (cont)
- Pod probe examples
- RestartPolicy examples
- InitContainers Pod resource management
- Pod security context
- Patterns for Composite Containers
- ReplicationController and ReplicaSet
- Working with ReplicationController, ReplicaSet
- Deployments
- Working with Deployments
- Kubernetes Lab: Workloads

Module 4: Scheduling and node management

- The Kubernetes Scheduler
- Pod priorities and preemption
- Assigning Pods to Nodes
- Assigning Pods to Nodes – Node affinities Assigning Pods to Nodes – Pod affinities
- Taints and tolerations
- Managing nodes
- Kubernetes Lab: Scheduling

Module 5: Accessing the applications

- Services
- Service types
- Working with Services
- Working with Services
- Ingress
- Ingress definition
- Working with Ingress
- Network Policies
- Network Policy example
- Kubernetes Lab: Accessing Applications

Module 6: **Persistent storage in Kubernetes**

- Volumes Volume example Volume types
- Persistent Volumes
- Persistent Volume example
- Dynamic PVC provisioning
- Secrets
- Using Secrets as environmental variables
- Using Secrets as volumes
- ConfigMaps
- Kubernetes Lab: Persistent Storage

Module 7: **Kubernetes Special Workloads**

- StatefulSets StatefulSets - Limitations
- StatefulSet example
- StatefulSet example with PVC
- Jobs, CronJobs
- Jobs example
- CronJobs example
- DaemonSets
- Kubernetes Lab: Special workloads

Module 8: **Logging, monitoring and troubleshooting**

- Logging architecture
- Monitoring
- Troubleshooting
- Kubernetes Lab: Logging and Monitoring

Module 9: **Installing and upgrading Kubernetes**

- Picking the right solution
- One node Kubernetes install
- Kubernetes universal installer
- Install using kubeadm on CentOS
- Upgrading Kubernetes
- Kubernetes Networking Kubernetes
- Lab:Upgrading Kubernetes

Appendix: **Application containers**

- Application containers
- Containers on Linux
- Container runtime

PART II. Kubernetes and Container-based Application Security with CKS exam.prep.

Module 10: User and authorization management

- Users and service accounts in Kubernetes
- Authenticating users
- Managing authorizations with RBAC

Module 11: Supply chain security

- Vulnerability checking for images
- Image validation in Kubernetes
- Reducing image footprint
- Secure image registries

Module 12: Validating cluster setup and penetration testing

- Use CIS benchmark to review the security configuration of Kubernetes components
- Modify the cluster components' configuration to match the CIS Benchmark
- Penetration testing Kubernetes for known vulnerabilities

Module 13: System hardening

- Use kernel hardening tools
- Setup appropriate OS level security domains
- Container runtime sandboxes
- Limit network access

Module 14: Monitoring and logging

- Configure Kubernetes audit logs
- Configure Audit Policies
- Monitor applications behaviour with Falco