

# Koenig Crafted – Kubernetes Boot Camp (CKA + CKAD + CKS)

**Duration: 10days (8hrs/day)**

**Prerequisites:**

- Basic knowledge of Linux Server Administration.
- Basic knowledge of Containers

**Course Objective:** This comprehensive course equips you with the knowledge and skills to confidently manage and orchestrate containerized applications using Kubernetes. Through hands-on experience, you will gain mastery over core concepts, navigate the architecture, perform installation and configuration, manage resources and deployments, secure your environment, and monitor your Kubernetes cluster effectively. By the end, you'll be prepared to architect, implement, and maintain reliable and scalable containerized applications in production environments.

**Kubernetes Version:** Latest

**Lab Requirement:** Koenig-DC

## Module 1 – Core Concepts

Overview of Container Orchestration

Introduction to Kubernetes

Understanding Kubernetes Architecture

## Module 2 – Installation, Configuration & Validation

Design a Kubernetes Cluster

**Lab:** Installation of Kubernetes 1-Master and 2-Nodes Cluster

**Lab:** Choose a Network Solution and Configure

**Lab:** Verify Installation with Kubectl command

## Module 3 – Creating Kubernetes Resources

Understanding Pods, Labels & Selectors

**Lab:** Deploying Applications as a Pod

**Lab:** Managing Labels & Selector

Understanding Replication Controller & Replica Set

**Lab:** Deploying Replication Controller & Replica Set

Understanding Services – ClusterIP, NodePort & LoadBalancer

**Lab:** Creating & Managing Service

Understanding Daemon Sets

**Lab:** Deploying Applications as Daemon Sets

## **Module 4 - Scheduling**

Manual Scheduling of Pods

Taint and Tolerations

**Lab:** Using Manual Scheduling or Taints and Tolerations

## **Module 5 - Application Lifecycle Management**

Overview of Deployment

Deployment Strategies – Blue/Green & Canary

**Lab:** Deploying Applications as Deployment

**Lab:** Implementing Deployment Strategies on Deployments

## **Module 6 - Environment Variable**

Plain Key

Config Map

Secret

**Lab:** Using Plain Keys, Config Map & Generic Secret as Environment Variables

**Lab:** Mount Environment Variable as Volumes

## **Module 7 – Storage**

Understanding Volume Management in K8s

Types of Volumes Provisioning

Persistent Volumes

Persistent Volume Claim

**Lab:** Using PV & PVC to attach Persistent Volume to a Pod as HostPath

## Understanding Storage Class

### **Module 8 – StatefulSet**

Introduction to StatefulSet

Use cases of StatefulSet

Manage StatefulSet

Storage in StatefulSet

**Lab:** Deploying and Managing Stateful Sets

**Lab:** Creating Persistent Storage in Stateful Sets

Headless Service

### **Module 9 – Security**

Understanding Kubernetes Authentication

**Lab:** Creating and Managing Users in Kubernetes

**Lab:** Creating Service Accounts

Understanding Role, ClusterRole, RoleBinding & ClusterRoleBinding

**Lab:** Managing Roles and Role Binding

**Lab:** Managing Cluster Role and Cluster Role Binding

Understanding Security Context

**Lab:** Adding Security Context to Pod to enable ping

### **Module 10 – Cluster Maintenance**

Understanding OS Upgrade

**Lab:** Upgrade a Kubernetes Cluster Version

Static Pod

**Lab:** Deploying Pods as Static Pod

**Lab:** ETCD Backup

CronJob

**Lab:** Deploying Pod as CronJob

## **Module 11 – Logging and Monitoring**

Understand how to Monitor Application and Cluster Components

**Lab:** Understand how to Read Application & Cluster Component Logs

**Lab:** Deploying Prometheus & Grafana to Monitor K8s Cluster

## **Module 12 – Networking in Kubernetes**

Understand Basics of Kubernetes Networking

Understand CNI overview

Understand Pod Networking Concepts

CoreDNS overview of K8s

Understanding Ingress

**Lab:** Configure and Manage Ingress Rule

Understanding Namespace & Use-Cases

**Lab:** Creating Namespace & Deploying K8s resources in Different Namespace

Metal Load Balancer

**Lab:** Deploying Metal Load Balancer

## **Module 13 – Multi Container Pod Design**

Understanding Multi-Container Pods

Creating Multi-Container Pods

**Lab:** Sidecar Pattern

**Lab:** Deploying Init Container

**Lab:** Ambassador Pattern

**Lab:** Adapter Pattern

## **Module 14 – Helm Package Manager**

Introduction to Helm

Work with Helm Charts

Create Helm Charts

**Lab:** Installing Helm Package Manager

Upgrade and Downgrade Helm Charts

**Lab:** Deploying Kubernetes Resources using Helm Package Manager

## **Module 15 – Building Docker Images**

Introduction to Dockerfile

Dockerfile Instructions

**Lab:** Building Container images using Dockerfile

Build Image Push Image to Centralized Registry

**Lab:** Pushing Container Image to a Public & Private Registry

## **Module 16 – Readiness and Liveness Probe**

Introduction to Readiness and Liveness Probe

Implement Readiness and Liveness in Pod

**Lab:** Creating Liveness and Readiness Probe for Pod

## **Module 17 – Troubleshooting**

Ways to Troubleshoot ETCD Failure

Ways to Troubleshoot Kubelet Failure

Ways to Troubleshoot Container Runtime Failure

Ways to Troubleshoot Scheduler Failure

## **Module 18 – Cluster Hardening**

Use CIS Benchmark to Review the Security Configuration of Kubernetes Components

**Lab:** Perform Security Benchmark checks using CIS-CAT Lite and Kube-Bench Tool

Public Key Infrastructure (PKI) – Certificate Authority (CA)

**Lab:** Find Certificates

**Lab:** Implementing Network Policies on Pods

Minimize Use of, and Access to, GUI Elements

**Lab:** Install Kubernetes Dashboard

**Lab:** Verify Platform Binaries - Theory and Hashes

Exercise Caution in Using Service Accounts e.g., Disable Defaults, Minimize Permissions on Newly Created Ones

**Lab:** Create User and assign RBAC (Role Based Access Control)

**Lab:** Disable Automount Service Account Token and Anonymous Access

**Lab:** Node Restriction Admission Controller

**Lab:** Update Kubernetes Frequently

## **Module 19 – Minimize Microservice Vulnerabilities**

**Lab:** Managing Secrets

**Lab:** Encrypting Secret in ETCD

Setup Appropriate OS Level Security Domains e.g. Using PSP, OPA, Security Contexts

**Lab:** Implementing Security Context in Pods and Containers

**Lab:** Creating privileged containers using security context

**Lab:** Disable Privilege Escalation

Pod Security Policy

Container Runtime Sandboxes

Open Container Initiative (OCI)

Kata Containers - Sandbox

Use Gvisor

**Lab:** Contact the Linux Kernel of worker node From Inside a Container

**Lab:** Implementing Gvisor on pods

**Lab:** Custom Security Policies using OPA Gatekeeper

## **Module 20 – Supply Chain Security**

Minimize Base Image Footprint Use Static Analysis of User Workloads (e.g. Kubernetes Resources, Docker Files)

**Lab:** Static Analysis with Kubesec

**Lab:** Static Analysis with OPA Conftest

Scan Images for Known Vulnerabilities

**Lab:** Checking Image Vulnerabilities with Trivy

Secure Supply Chain

**Lab:** Whitelist Some Registering Using OPA

ImagePolicyWebhook

## **Module 21 – Monitoring, Logging and Runtime Security**

Perform Behavioral Analytics of Syscall Process and File Activities at the Host and Container Level to Detect Malicious Activities

Kernel vs User Space

**Lab:** Using Strace command to trace Syscall

Falco

Immutability of Containers at Runtime

**Lab:** Implementing Immutability on Containers

**Lab:** Enforce Read-Only Root Filesystem

Use Audit Logs to Monitor Access

**Lab:** Configure API Server To Store Audit Logs

**Lab:** Restrict Amount of Audit Logs to Collect

## **Module 22 – System Hardening**

Kernel Hardening Tools

Linux Kernel Isolation

**Lab:** AppArmor

**Lab:** Kubernetes with AppArmor

**Lab:** Seccomp with Kubernetes

Minimize OS Footprint - Reduce Attack Surface

**Lab:** Reduce Attack Surface