

## **KBS-505: Kubernetes, Helm and Istio Admin. With CKA & CKAD exam.prep.**

**Course Length:** 3 days Kubernetes + 1 day Helm + 1 day Istio, 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 and third part enhances the delegates' knowledge with more advanced Kubernetes features and extensions such as Helm and Istio.

This course doesn't only prepare delegates for the daily administration of Docker & Kubernetes systems but also for the official [Certified Kubernetes Administrator \(CKA\)](#) and [Certified Kubernetes Application Developer \(CKAD\) exams](#) 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 with CKA and CKAD exam.prep.**

##### **Module 1: Kubernetes Introduction**

- Cloud computing in general
- Cloud types
- Cloud native computing
- Container orchestration
- Kubernetes
- Kubernetes concepts
- Kubernetes objects categories
- Kubernetes architecture
- Kubernetes master
- Kubernetes node
- Kubernetes Lab 1

## Module 2: Accessing Kubernetes

- Accessing the Kubernetes cluster
- Controlling access to the API
- Authorization
- Role Based Access Control
- Roles and ClusterRoles
- Role bindings
- Kubernetes Lab 2

## Module 3: Kubernetes Workloads

- The pod
- Our first Pod
- Operations on pods
- Pod Lifecycle
- 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
- Jobs, CronJobs
- Jobs example
- CronJobs example
- DaemonSets
- Kubernetes Lab 3

## 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 – node affinities
- Assigning Pods to Nodes – Pod affinities
- Assigning Pods to Nodes – pod affinities
- Taints and tolerations
- Managing nodes
- Kubernetes Lab 4

## **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 5

## **Module 6: Persistent storage in Kubernetes**

- Volumes
- Volume example
- Volume types
- Persistent Volumes
- Persistent Volume example
- Secrets
- Using Secrets as environmental variables
- Using Secrets as volumes
- ConfigMaps
- Kubernetes Lab 6

## **Module 7: Logging, monitoring and troubleshooting**

- Logging architecture
- Monitoring
- Troubleshooting
- Kubernetes Lab 7

## **Module 8: Installing Kubernetes**

- Picking the right solution.
- One node Kubernetes install
- Kubernetes universal installer
- Install using kubeadm on CentOS
- Kubernetes Networking
- Kubernetes Lab 8

## **Appendix: Application containers**

- Application containers
- Containers on linux
- Container runtime

## **PART II. 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 III: Istio Service Mesh**

### **Module 7: Introduction to Istio**

- Service mesh
- What is Istio?
- Istio features
- Platform support
- Istio architecture
- Istio architecture
- Istio architecture – Data plane
- Istio Architecture - Control plane
- Istio's design goals

### **Module 8: Installing Istio in Kubernetes**

- Installing Istio on Kubernetes
- Injecting the sidecar container
- Lab:Installing Istio on Kubernetes

### **Module 9: Istio Traffic Management**

- Traffic management
- Traffic routing concepts
- Request routing
- Discovery and load balancing
- Handling failures
- Rule configuration
- Virtual Services
- Virtual Services - examples
- Virtual Services – Timeouts and retries
- Virtual Services – Injecting faults
- Destination rules
- Destination rules – Circuit breakers
- Service entries
- Service entries - example
- Gateways
- Gateways - example
- Sidecars
- Lab:Traffic management

## **Module 10: Security**

- Security
- Security - architecture
- Security – Identity
- Service Authentication
- Authentication policies - scope
- Authentication policies (cont)
- Authorization
- Service roles
- Service role binding
- Lab:Security

## **Module 11: Policies and telemetry**

- Policies and telemetry
- Reliability and latency
- Attributes
- Configuration model
- Handlers
- Instances
- Rules
- Lab: Policies and telemetry