

## **KBS-104: Kubernetes Administration and Helm with CKA & CKAD exam.prep.**

**Course Length:** 3 days Kubernetes + 1 day Helm, 4 days altogether

### **Course Description:**

Kubernetes is the leading open-source system for automating deployment, scaling and management of containerized applications.

This course 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 and its Logging, Monitoring and Troubleshooting facilities.

The second part deals with Helm, the Kubernetes Package Manager.

This course doesn't only deal with the daily administration of Kubernetes based systems but also prepare delegates 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 cloud and data center 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. KBS-103 Kubernetes Admin with CKA & CKAD exam.prep.**

### **Module 1: Introduction**

- Cloud computing in general
- Cloud types
- Cloud native computing
- Application containers
- Containers on Linux
- Container runtime
- Container orchestration
- Kubernetes
- Kubernetes concepts
- Kubernetes objects categories
- Kubernetes architecture
- Kubernetes master
- Kubernetes node
- Lab 1

## **Module 2: Installing Kubernetes**

- Picking the right solution.
- One node Kubernetes install
- Kubernetes universal installer
- Install using kubeadm on CentOS
- Kubernetes Networking
- Lab 2

## **Module 3: Accessing Kubernetes**

- Accessing the Kubernetes cluster
- Controlling access to the API
- Authorization
- Role Based Access Control
- Roles and ClusterRoles
- Role bindings
- Lab 3

## **Module 4: Kubernetes Workloads**

- The pod
- RestartPolicy examples
- InitContainers
- Our first Pod
- Operations on pods
- Replication Controller
- Working with Replication Controller
- Deployments
- Working with Deployments
- Jobs, CronJobs
- Jobs example
- CronJobs example
- DaemonSets
- Lab 4

## **Module 5: Scheduling and node management**

- The Kubernetes Scheduler
- Assigning Pods to Nodes
- Assigning Pods to Nodes – node affinities
- Assigning Pods to Nodes – Pod affinities
- Taints and tolerations
- Managing nodes
- Lab 5

## **Module 6: Accessing the applications**

- Services
- Service types
- Working with Services

- Working with Services
- Ingress
- Ingress definition
- Working with Ingress
- Network Policies
- Network Policy example
- Lab 6

## **Module 7: Persistent storage in Kubernetes**

- Volumes
- Volume types
- Persistent Volumes
- Secrets
- Using Secrets as environmental variables
- Using Secrets as volumes
- ConfigMaps
- Lab 7

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

- Logging architecture
- Monitoring
- Troubleshooting

## **PART II. Helm Package Manager**

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

- What is Helm?
- Helm concepts
- Helm v2 (legacy) components
- Helm v3 components
- Installing Helm
- Helm Lab: Installing Helm

### **Module 2: Using Helm**

- Generic options and help
- Working with repositories
- Finding charts
- Installing a release
- List releases
- Upgrade/rollback releases
- Uninstalling releases
- Helm Lab: Using Helm

### **Module 3: Helm charts**

- Introduction to charts
- The structure The Chart.yaml File
- The components of a Chart
- Chart dependencies
- Chart dependencies (cont.)

- Managing Charts with helm
- Helm Lab: Working with charts

#### **Module 4: Chart Templates**

- Writing Templates Templates and Values
- Dependencies and values Dependencies and values
- Chart lifecycle hooks Functions and pipelines
- Flow control
- Variables
- Named templates
- Helm Lab: Writing templates

#### **Module 5: Helm plugins**

- Building plugins
- Helm Lab: Helm plugins