

KBS-405: Docker & Kubernetes Admin with CKA & CKAD exam.prep.

Course Length: 2 days Docker + 3 Days Kubernetes, 5 days altogether

Course Description:

Docker is the de-facto standard of Linux and Windows containers and one of the main building blocks of modern cloud-native, microservices based IT systems. And Kubernetes is the de-facto system for container orchestration, e.g. automating the deployment, scaling and management of microservices-based, containerized applications.

This course first introduces students to the main concepts of containers generally and Docker specifically and covers all its core features including: installing and basic management of Docker containers, Docker networking and storage, managing images and using Dockerfile to create and manage custom images, Docker Registry, Integrating Docker into Eclipse IDE as well as connecting multiple containers.

The second part of the 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 as well as its Logging, Monitoring and Troubleshooting facilities.

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 Docker and Kubernetes in enterprise and cloud environments.

Prerequisites: Proficiency with the Linux CLI. A broad understanding of Linux system administration.

Detailed Course Outline

PART I. DCK-102 Docker administration

Module 1: INTRODUCTION TO CLOUD AND CONTAINERS

- Cloud computing in general
- Cloud types
- Cloud native computing
- Application containers
- Containers on Linux
- Container runtime
- Docker
- Docker Ecosystem
- Docker Ecosystem
- Lab 1

Module 2: INSTALLING DOCKER

- Distribution Packages
- Packages from Docker
- Docker-machine
- Docker Daemon Configuration
- Lab 2

Module 3: MANAGING CONTAINERS

- Docker - container run
- Docker – container ps
- Docker – container attach
- Docker – container signal
- Docker - accessing container FS
- Lab 3

Module 4: DOCKER NETWORK - PORT PUBLISHING

- Port publishing
- Lab 4

Module 5: DOCKER STORAGE - IMAGES

- Docker storage
- Docker storage - overlay FS
- Docker storage - image layers
- Docker images - commit
- Docker images - cli
- Docker images - work with images
- Lab 5

Module 6: BUILDING IMAGES

- Docker images - Dockerfile
- Docker images - Dockerfile syntax
- Docker images - Dockerfile syntax
- Docker images - Dockerfile syntax
- Lab 6

Module 7: DOCKER REGISTRY

- Docker registry
- Docker registry CLI
- Docker local registry setup
- Lab 7

Module 8: INTEGRATING DOCKER INTO ECLIPSE IDE

- Eclipse IDE for Developers
- Creating sample code - java
- Building and Testing
- Lab 8

Module 9: DOCKER STORAGE - VOLUME

- Docker storage - volume
- Docker storage - volume management
- Docker storage - shared volume
- Docker storage - shared volume examples
- Lab 9

Module 10: CONNECTING CONTAINERS

- Connecting Containers
- Lab 10

PART II. KBS-103 Kubernetes administration with CKA & CKAD exam.prep.

Module 11: 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 12: 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 13: 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 14: 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 15: 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 16: 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 17: Kubernetes Special Workloads

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

Module 18: Logging, monitoring and troubleshooting

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

Module 19: 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