

Kubernetes Administration and CKA Certification

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: TP3339

Duration: 5 days

Overview

Course Description

This Kubernetes (K8s) course teaches attendees how to create, manage, and deploy a K8s application. It is also excellent preparation for the Certified Kubernetes Administrator (CKA) certification. The CKA exam is a hands-on, performance-based exam administered by the Linux foundation that tests the ability to administer Kubernetes clusters.

Skills Gained

- Create a Kubernetes cluster, including installing the necessary components, configuring the cluster, and adding nodes
- Deploy applications to Kubernetes by creating pods, services, and storage
- Manage Kubernetes resources with scaling, autoscaling, and updating applications
- Secure Kubernetes by configuring authentication, authorization, and admission control.
- Troubleshoot Kubernetes by monitoring logs, diagnosing application failures, and troubleshooting network access

Prerequisites

- Understand Linux administration skills
- Comfortable using the command line
- Be able to edit files using a command-line text editor
- Experience with containers, and networking

Course Outline

- Introduction
- Kubernetes Concepts
 - Kubernetes Basics
 - Container Orchestration
 - Kubernetes Architecture
 - Kubernetes Concepts
 - Cluster and Namespace
 - Control Plane
 - Nodes
 - Pods
 - Storage
 - Services
 - Objects
 - Object Specifications
 - Labels & Selectors
 - Tools (kubeadm, kubectl)
 - Essential K8S commands
 - Essential Linux commands
- Installation and Configuration
 - Installing kubeadm
 - Obtaining Nodes
 - Installing Kubelet
 - Installing kubectl
 - Initializing the control plane
 - Setup kubeconfig file
 - Joining nodes to the cluster

- Verifying the cluster
- Cluster Administration
 - Cluster Components
 - Installation and Setup
 - Security
 - Resource management
 - Scheduling
 - Monitoring
 - Logging
 - Scaling
 - Autoscaling
 - High Availability
 - Upgrading the cluster version
 - Backup and restore the etcd store
- Workloads
 - Application Images
 - Deploying Pods
 - Deploying Workloads
 - Self-healing applications
 - Deployments
 - Deployment States
 - Replica Sets
 - Daemon Sets
 - Scaling a workload
 - Autoscaling a workload
 - Deleting workloads
 - Updating a workload image
 - Update strategies
 - Rolling updates
 - Rolling back an update
- Scheduling
 - Pod Scheduling
 - Resource usage basics
 - Setting Resource Limits

- Resource Requests
- Resource Quotas
- Optimizing Resource Usage
- Node Affinity
- Pod Affinity
- Taints
- Tolerations
- Services
 - Available Service Types
 - ClusterIP Service
 - NodePort Service
 - LoadBalancer Service
 - Creating services
 - Accessing workloads through services
- Networking
 - K8s Networking Overview
 - Ingress Controller
 - Ingress Resources
 - Resource Discovery
 - CoreDNS
 - Configuring CoreDNS
 - Port forwarding
- Storage
 - Storage Classes
 - EmptyDir
 - HostPath
 - Persistent volumes
 - Persistent volume claims
 - NFS share volumes
 - Cloud storage volumes
 - Access Modes
 - Reclaim Policies
 - Configuring persistent storage for workloads
 - Secrets

- Config Maps
- Security
 - Securing the API
 - Securing ETCD
 - Securing Worker nodes
 - Securing images
 - Authentication
 - Authorization
 - Admission Controller
 - Securing cluster access
 - Kubectl config
 - RBAC Role Based Access Control
 - Pod policies
 - Network Policies
- Troubleshooting
 - Logs and log access
 - Standard output and error logs
 - Workload monitoring
 - Application failure
 - Network access troubleshooting
 - Troubleshooting cluster components
- Conclusion