# Certified Kubernetes Application Developer (CKAD)

## Course Duration: 40 Hours (5 Days)

## Overview

The Certified Kubernetes Application Developer (CKAD) course is designed for individuals who want to demonstrate their skills in designing, building, configuring, and Exposing cloud-native applications for Kubernetes. The course goes through a series of modules that cover all aspects needed to become proficient in working with Kubernetes, including core concepts, configuration, multi-container pods, observability, pod design, services & networking, and state persistence. Throughout the course, students gain hands-on experience with Kubernetes security training, learning to secure pods and clusters. They also acquire the competencies expected from a Kubernetes administrator training, such as Managing applications using Kubernetes primitives and ensuring their applications are properly orchestrated and maintained. By the end of the course, learners will have a solid understanding of how to work with Kubernetes in a practical way, which will help them in developing, deploying, and maintaining scalable and highly available applications on the Kubernetes platform.

## Audience Profile

The Certified Kubernetes Application Developer (CKAD) course is designed for professionals seeking expertise in building, deploying, and configuring applications in Kubernetes.

- Software Developers and Engineers familiar with containerization concepts
- DevOps Engineers focusing on Continuous Integration and Continuous Deployment (CI/CD) workflows
- Cloud Engineers working with Kubernetes on cloud platforms like AWS, GCP, and Azure
- Systems Administrators looking to manage containerized applications
- IT Professionals seeking to validate their Kubernetes skills with a certification
- Application Architects designing scalable and fault-tolerant applications on Kubernetes
- Site Reliability Engineers (SREs) ensuring application reliability and performance in a Kubernetes environment
- Technical Leads overseeing development teams working with Kubernetes
- Product Managers who need to understand the technical aspects of Kubernetes to manage product lifecycles effectively
- Technical Support Staff aiding for Kubernetes-based applications and infrastructure

## Course Syllabus

### Module 1 – Core Concepts

- Overview of Container Orchestra on
- 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 Solu on and Configure
- Lab: Verify Installation on with Kubectl command
- Metal Load Balancer
- Lab: Deploying Metal Load Balancer

## Module 3 – Creating Kubernetes Resources

- Understanding Pods, Labels & Selectors
- Lab: Deploying Applications as a Pod
- Lab: Managing Labels & Selector
- Understanding Replica on Controller & Replica Set
- Lab: Deploying Replica on Controller & Replica Set
- Understanding Services – Cluster IP, Node Port & Load Balancer
- Lab: Creating & Managing ClusterIP, Node Port and Load Balancer
- 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
- Node Selector
- Lab: Using Node Selector to Deploy Pods
- Node Affinity
- Lab: Using Node Affinity to Deploy Pods

## Module 5 - Application Lifecycle Management

- Overview of Deployment
- Deployment Strategies – Blue/Green & Canary
- Lab: Deploying Applications as Deployment
- Lab: Implementing Blue-Green Deployment Strategy

## 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 a ach Persistent Volume to a Pod as Host Path
- Understanding Storage Class

## Module 8 – Security

- Understanding Namespace & Use-Cases
- Lab: Creating Namespace & Deploying K8s resources in Different Namespace
- Lab: Creating Service Accounts
- Understanding Role, Cluster Role, Role Binding & Cluster Role Binding
- Lab: Managing Cluster Role and Cluster Role Binding
- Understanding Security Context
- Lab: Adding Security Context to Pod to enable ping

## Module 9 – Networking in Kubernetes

- Understand Basics of Kubernetes Networking
- Understand CNI overview
- Understand Pod Networking Concepts
- Lab: Controlling Pod Communication using Network Policies
- CoreDNS overview of K8s
- Understanding Ingress
- Lab: Configure and Manage Ingress Rule

## Module 10 – 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
- Lab: Headless Service

## Module 11 – 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 12 – Logging and Monitoring

- Understand how to Monitor Applica on and Cluster Components
- Lab: Understand how to Read Applica on & Cluster Component Logs
- Lab: Deploying Prometheus & Grafana to Monitor K8s Cluster

## Module 13 – Multi Container Pod Design & Jobs

- Understanding Mul-Container Pods
- Creating Mul-Container Pods
- Lab: Sidecar Pattern
- Lab: Deploying Init Container
- Lab: Ambassador Pattern
- Lab: Adapter Pattern
- CronJob
- Lab: Deploying Pod as a CronJob

## 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 Overview
- Lab: Building Container images using Dockerfile
- Build Image Push Image to Centralized Registry
- Lab: Pushing Container Image to a Public Registry