Document Generated: 11/05/2026

Learning Style: Virtual Classroom

Technology: Linux Foundation

Difficulty: Intermediate

Course Duration: 4 Days

Kubernetes for App Developers (LFD459)



About this course:

Kubernetes is a high-velocity open source orchestration tool to deploy, scale, and update containerized applications. This course will teach you how to containerize, host, deploy, and configure an application in a multi-node cluster.

The topics covered are directly aligned with the knowledge domains tested by the Cloud Native Computing Foundation Certified Kubernetes Application Developer (CKAD) Program, and will substantially increase students' ability to become certified.

This course will teach you how to containerize, host, deploy, and configure an application in a multi-node cluster. Starting with a simple Python script, you will define application resources and use core primitives to build, monitor and

troubleshoot scalable applications in Kubernetes. Working with network plugins, security and cloud storage, you will be exposed to many of the features needed to deploy an application in a production environment.

This course is designed to be vendor- and distribution-neutral, so you will be able to apply these concepts universally.

You will have access to the course for a full year from the date of purchase (or availability, for preorders), regardless of how quickly you complete the course. You can expect the course to take 30-35 hours to complete (although the course is self-paced, so you can move as quickly or as slowly as you like).

The average salary of a Kubernetes Engineer is **$143,283** per year.

Course Objective:

In this course you'll learn how to:

- Containerize and deploy a new Python script
- Configure the deployment with ConfigMaps, Secrets and SecurityContexts
- Understand multi-container pod design
- Configure probes for pod health
- Update and roll back an application
- Implement services and NetworkPolicies
- Use PersistentVolumeClaims for state persistence
- And more

Audience:

- App Developers
- Kubernetes Engineers
- Linyx Developers

Prerequisite:

To get the most out of this course, you should have basic Linux command line and file editing skills and be familiar with using a programming language (such as Python, Node.js, Go). A knowledge of Cloud Native application concepts and architectures.

Course Outline:

Introduction

- Objectives
- Who You Are
- The Linux Foundation
- Linux Foundation Training
- Preparing Your System

- Course Registration
- Labs

Kubernetes Architecture

- What Is Kubernetes?
- Components of Kubernetes
- Challenges
- The Borg Heritage
- Kubernetes Architecture
- Terminology
- Master Node
- Minion (Worker) Nodes
- Pods
- Services
- Controllers
- Single IP per Pod
- Networking Setup
- CNI Network Configuration File
- Pod-to-Pod Communication
- Cloud Native Computing Foundation
- Resource Recommendations
- Labs

Build

- Container Options
- Containerizing an Application
- Hosting a Local Repository
- Creating a Deployment
- Running Commands in a Container
- Multi-Container Pod
- readinessProbe
- livenessProbe
- Testing
- Labs

Design

- Traditional Applications: Considerations
- Decoupled Resources
- Transience
- Flexible Framework
- Managing Resource Usage
- Multi-Container Pods
- Sidecar Container
- Adapter Container
- Ambassador
- Points to Ponder
- Labs

Deployment Configuration

- Volumes Overview
- Introducing Volumes
- Volume Spec
- Volume Types
- Shared Volume Example
- Persistent Volumes and Claims
- Persistent Volume
- Persistent Volume Claim
- Dynamic Provisioning
- Secrets
- Using Secrets via Environment Variables
- Mounting Secrets as Volumes
- Portable Data with ConfigMaps
- Using ConfigMaps
- Deployment Configuration Status
- Scaling and Rolling Updates
- Deployment Rollbacks
- Jobs
- Labs

Security

- Security Overview
- Accessing the API
- Authentication
- Authorization
- ABAC
- RBAC
- RBAC Process Overview
- Admission Controller
- Security Contexts
- Pod Security Policies
- Network Security Policies
- Network Security Policy Example
- Default Policy Example
- Labs

Exposing Applications

- Service Types
- Services Diagram
- Service Update Pattern
- Accessing an Application with a Service
- Service without a Selector
- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

- Ingress Resource
- Ingress Controller
- Labs

Troubleshooting

- Troubleshotting Overview
- Basic Troubleshooting Steps
- Ongoing (Constant) Change
- Basic Troubleshooting Flow: Pods
- Basic Troubleshooting Flow: Node and Security
- Basic Troubleshooting Flow: Agents
- Monitoring
- Logging Tools
- Monitoring Applications
- System and Agent Logs
- Conformance Testing
- More Resource
- Labs

Closing and Evaluation Survey

Credly Badge:



Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your Linkedin profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

Find Out More or See List Of Badges