

Kubernetes Application Developer (CKAD)

CODICE	DT0186
DURATA	4 gg
PREZZO	2.500,00 €
EXAM	

DESCRIZIONE

Il Corso Kubernetes Developers CKAD è progettato per insegnare come utilizzare Kubernetes per distribuire, scalare e gestire le applicazioni containerizzate. Gli obiettivi del corso includono la comprensione dell'architettura e dei componenti di Kubernetes, l'interazione con un cluster Kubernetes tramite kubectl, l'uso di manifesti YAML per definire e distribuire applicazioni, l'implementazione di servizi di rete per esporre le applicazioni, la gestione dello storage per i dati delle applicazioni, l'implementazione di best practice di sicurezza e accesso, l'integrazione di Kubernetes con i processi di Continuous Integration e Continuous Deployment (CI/CD) e, infine, la preparazione per l'esame di Certificazione Kubernetes Application Developer CKAD.

TARGET

Questo corso è rivolto agli ingegneri Kubernetes, agli ingegneri cloud e ad altri professionisti IT responsabili della creazione, distribuzione e configurazione di applicazioni cloud native con Kubernetes.

PREREQUISITI

Conoscenza di base runtime dei container e dell'architettura dei microservizi.

CONTENUTI

Kubernetes Architecture

- Introduction to Kubernetes Architecture
- Components of Kubernetes
- Challenges
- The Borg Heritage
- Kubernetes Architecture
- Terminology
- Control Plane Node
- Control Plane Node Components

- Worker Nodes
- Pods
- Services
- Operators
- Single IP per Pod
- Networking Setup
- API Call Flow
- CNI Network Configuration File
- Pod-to-Pod Communication
- Cloud Native Computing Foundation (CNCF)
- Resource Recommendations

Build

- Introduction to Build
- Build
- Container Options
- Container Runtime Interface (CRI)
- containerd
- CRI-O
- Docker
- rkt
- Containerizing an Application
- Rewrite Legacy Applications
- Creating the Dockerfile
- Hosting a Local Repository
- Creating a Deployment
- Running Commands in a Container
- Multi-Container Pod
- readinessProbe, liveness Probe, and startupProbe
- Testing
- Helm

Design

- Introduction to Design
- Design
- Traditional Applications – Considerations
- Decoupled Resources
- Transience
- Flexible Framework
- Managing Resource Usage

- Using Label Selectors
- Multi-Container Pods
- Sidecar, Adapter, Ambassador, and initContainers
- Custom Resource Definitions
- Points to Ponder
- Jobs

Deployment Configuration

- Introduction to Deployment Configuration
- Deployment Configuration
- Volumes Overview
- Introducing Volumes
- Volume Spec
- Volume Types
- Shared Volume Example
- Persistent Volumes and Claims
- Phases to Persistent Storage
- 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

Understanding Security

- Introduction to Understanding Security
- Security
- Accessing the API
- Authentication
- Authorization
- RBAC and the RBAC Process Overview
- Admission Controller
- Security Contexts
- Pod Security Policies (PSPs)
- Pod Security Standards

- Network Security Policies
- Network Security Policy Example
- Default Policy Example

Exposing Applications

- Introduction
- Introduction to Exposing Applications
- 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
- Service Mesh

Application Troubleshooting

- Troubleshooting
- 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

Attività Laboratoriali

- How to connect to lab
- Work with the cluster
- Imperative and declarative
- API
- Namespace
- Labels

- Running Pods
- Pod Lifecycle
- Implementing container probes
- System Pods and Controllers
- Create Deployment
- Create Deployment-Replicaset
- Update a Deployment
- Rolling back a Deployment
- Scaling a Deployment
- Running Multi-container Pods
- Running Sidecar Containers
- Running Adapter Containers
- Storage Static Provisioning
- Storage Dynamic Provisioning
- Environment Variables
- Secrets
- Private Container Registry
- ConfigMap
- Services
- Service Discovery
- Ingress
- Logging
- Troubleshooting Workloads
- Authentications
- Manage Authorizations
- kubeconfig and context
- RBAC