



Kubernetes Administrator (CKA)

CODICE	DT0185
DURATA	4 gg
PREZZO	2.500,00 €
EXAM	

DESCRIZIONE

Il corso Kubernetes Administrator CKA è strutturato per offrire ai partecipanti una profonda comprensione e abilità pratiche nell'utilizzo di Kubernetes. Il corso affronta vari aspetti come l'architettura di Kubernetes, l'installazione e configurazione di un cluster, la gestione di applicazioni containerizzate, la creazione di servizi di rete e storage, e l'utilizzo di strumenti di monitoraggio e di registrazione. Inoltre, si esplorano strategie di sicurezza e meccanismi di autoscaling. Il corso contribuisce alla preparazione dell'esame per la Certificazione Kubernetes Administrator (CKA).

PREREQUISTI

Competenze sistemiche e di networking di base.

CONTENUTI

Basics of Kubernetes

- Basics of Kubernetes
- What Is Kubernetes?
- Components of Kubernetes
- Challenges
- Other Solutions
- Borg Heritage
- Kubernetes Architecture
- Terminology
- Innovation
- User Community
- Tools
- Cloud Native Computing Foundation (CNCF)
- Resource Recommendations

Installation and Configuration

- Installation Tools
- Installing kubectl
- Using Google Kubernetes Engine (GKE)
- Using Minikube
- Installing with kubeadm
- kubeadm-upgrade
- Installing a Pod Network
- More Installation Tools
- Installation Considerations
- Main Deployment Configurations
- Compiling from Source

Kubernetes Architecture

- Main Components
- Control Plane Node
- Worker Nodes
- Kubelet
- Operators
- Service Operator
- Pods
- Rewrite Legacy Applications
- Containers
- Init Containers
- Component Review
- API Call Flow
- Node
- Single IP per Pod
- Container to Outside Path
- Services
- Networking Setup
- CNI Network Configuration File
- Pod-to-Pod Communication

Apis and Access

- APIs and Access
- API Access
- RESTful
- Checking Access
- Optimistic Concurrency
- Using Annotations

- Simple Pod
- Manage API Resources with kubectl
- Access from Outside the Cluster
- ~/.kube/config
- Namespaces
- Working with Namespaces
- API Resources with kubectl
- Additional Resource Methods
- Swagger and OpenAPI
- API Maturity

Api Objects

- API Objects
- Overview
- v1 API Group
- Discovering API Groups
- Deploying an Application
- DaemonSets
- StatefulSets
- Autoscaling
- Jobs
- RBAC

Managing State with Deployments

- Deployments
- Object Relationship
- Deployment Details
- Deployment Configuration Metadata
- Deployment Configuration Spec
- Deployment Configuration Pod Template
- Deployment Configuration Status
- Scaling and Rolling Updates
- Deployment Rollbacks
- Using DaemonSets
- Labels

Volumes and Data

- Introducing Volumes
- Volume Spec
- Volume Types

- Shared Volume Example
- Persistent Volumes and Claims
- Persistent Volume
- Persistent Volume Claim
- Dynamic Provisioning
- Using Rook for Storage Orchestration
- Secrets
- Using Secrets via Environment Variables
- Mounting Secrets as Volumes
- Portable Data with ConfigMaps
- Using ConfigMaps

Services

- Service Update Pattern
- Accessing an Application with a Service
- Service Types
- Service Types (Continued)
- Services Diagram
- Overall Network View
- Local Proxy for Development
- DNS
- Verifying DNS Registration

Helm

- Helm
- Deploying Complex Applications
- Helm v3
- Chart Contents
- Templates
- Chart Repositories and Hub
- Deploying a Chart

Ingress

- Ingress Controller
- nginx
- Google Load Balancer Controller (GLBC)
- Ingress API Resources
- Deploying the Ingress Controller
- Creating an Ingress Rule
- Multiple Rules

- Intelligent Connected Proxies

Scheduling

- kube-scheduler
- Filtering (Predicates)
- Scoring (Priorities)
- Scheduling Policies
- Pod Specification
- Specifying the Node Label
- Scheduler Profiles
- Pod Affinity Rules
- podAffinity Example
- podAntiAffinity Example
- Node Affinity Rules
- Node Affinity Example
- Taints
- Tolerations
- Custom Scheduler

Logging and Troubleshooting

- Basic Troubleshooting Steps
- Ephemeral Containers
- Cluster Start Sequence
- Monitoring
- Using krew
- Managing Plugins
- Sniffing Traffic With Wireshark
- Logging Tools
- Understanding Fluentd
- Understanding Prometheus and Grafana
- More Resources

Custom Resource Definitions

- Configuration Example
- New Object Configuration
- Optional Hooks
- Understanding Aggregated APIs

Security

- Cloud Security Considerations

- Accessing the API
- Authentication
- Authorization
- RBAC and Webhook Modes
- Admission Controller
- Security Contexts
- Pod Security Policies
- Network Security Policies
- Network Security Policy Example
- Default Policy Example

High Availability

- High Availability
- Cluster High Availability
- Collocated Databases
- Non-Collocated Databases

Attività Laboratoriali

- How to connect to lab
- Install Kubernetes
- Work with the cluster
- Imperative and declarative
- API
- API Discovery
- Anatomy API Request
- Namespace
- Labels
- Running Pods
- Running Multi-container Pods
- Pod Lifecycle
- Implementing container probes
- System Pods and Controllers
- Create Deployment
- Create Deployment-Replicaset
- Update a Deployment
- Rolling back Deployment
- Scaling a Deployment
- Creating a Daemonset
- Executing Jobs and CronJobs
- Storage Static Provisioning

- Storage Dynamic Provisioning
- Environment Variables
- Secrets
- Private Container Registry
- ConfigMap
- Scheduling
- Scheduling with Affinity and Antiaffinity
- Node Cordoning
- Calico CNI Plugin
- Configure CoreDNS
- Services
- Service Discovery
- Ingress
- ETCD
- Upgrade Kubernetes
- Logging
- JSONpath
- Monitoring with Metric Server
- Troubleshooting Node failures
- Troubleshooting Control Plane failures
- Troubleshooting Workloads
- Authentications
- Manage Authorizations
- PKI on controller
- Create a certificate for a new user
- kubeconfig and context
- RBAC