Kubernetes – Fundamentals

Description

Is your team beginning to use Kubernetes for container orchestration? Do you need guidelines on how to start transforming your organization with Kubernetes and cloud native patterns? Would you like to simplify software container orchestration and find a way to grow your use of Kubernetes without adding infrastructure complexity? Then this is the course for you!
In this course, we'll discuss some of Kubernetes' basic concepts and talk about the architecture of the system, the problems it solves, and the model that it uses to handle containerized deployments and scaling and many more.

Classroom Registration Price (CHF)
2300
Virtual Classroom Registration Price (CHF)
2150
Course Content
The course is divided into several parts. These can be rearranged differently depending on the course's progression. Many practical exercises are offered to illustrate the different concepts. **Useful concepts:**

- Containers
- The Docker platform
- Microservice architecture
- Cloud native application
- DevOps
- The Docker platform

The project:

- Introduction
- History A flagship project of CNCF

Overview of main elements:

- Cluster
- Pod
- Deployment
- Service Labels and annotations
- Kubectl for communicating with a cluster
- Hands-on exercises: Installing kubectl

Architecture:

- Different types of nodes
- Processes
- Context
- Different interfaces of Kubernetes

Installation:

- Development/test cluster
- Production cluster
- Setting up on a cloud provider
- Manual setup
- Hands-on exercises: Setting up a cluster with kubeadm

Resources: Pod:

- Usage
- Example
- Lifecycle
- Scheduling
- Hands-on exercises: Launching a simple Pod and launching a Pod with multiple containers

Focus on kubectl:

- Imperative/declarative approach
- Highly used options
- Usage of plugins
- Some essential tools

Resources: Service:

- Usage
- Different types
- Lifecycle Demos
- Hands-on exercises: Creating a ClusterIP type service, creating a NodePort type service, and creating a LoadBalancer type service

Resources: Deployment:

- Usage
- Example specification
- ReplicaSet
- Update strategies
- Rolling update and rollback
- Hands-on exercises: Creating a Deployment and updating

Resources: Namespace:

- Introduction
- Creation
- Usage with a Context
- Quotas
- Hands-on exercises: Creating and using Namespaces and using quotas

Putting into practice:

- Deploying an application using previous elements

Resources: ConfigMap:

- Usage

- Different creation options
- Usage in a Pod
- Hands-on exercises: Using a ConfigMap for nginx server configuration and updating a ConfigMap

Resources: Secrets:

- Usage
- Different types (generic, docker registry, TLS)
- Demos
- Hands-on exercises: Using a Secret for connecting to an external database

Putting into practice:

- Deploying an Elastic stack using previous elements

Users and access rights:

- API server requests
- Authentication
- Authorization
- Demo
- Hands-on exercises: Defining a user and a group via x509 certification and setting access rights

Management interface:

- Introduction to the Kubernetes dashboard
- Other open source solutions
- Hands-on exercises: Deploying and using the dashboard

Resources: DaemonSet:

- Usage
- Example for log collection
- Illustration with a network plugin
- Hands-on exercises: Creating a DaemonSet

Resources: Ingress:

- Introduction
- Route by domain name or HTTP request
- -on exercises: Exposing an application via an Ingress

Stateful application:

- The notion of Volume
- Different types (emptyDir, hostPath, Persistent Volume, Persistent Volume Claim)
- StatefulSet
- Hands-on exercises: Using different types of Volume for persistence of an application

Helm:

- Introduction
- Deploying applications packaged in Helm charts
- Creating a Helm chart
- Hands-on exercises: Deploying the TICK stack and packaging in a Helm chart.

Documentation

- Digital courseware included

Participant profiles

- Application developers
- Cloud solution architects
- DevOps engineers
- IT managers

Prerequisites

- Good knowledge of Linux Command Line Interface (CLI)
- Basic knowledge of Docker container is recommended

Objectives

- To know its role and positioning in the ecosystem
- To have a global vision of the architecture
- To know the different processes used
- To know how to use the kubectl client
- To know the different primitives and how to use them
- To know the different options for installing an environment and development
- To know the different tools for setting up a production environment
- To have a working knowledge of application deployment
- To Understand user management and access rights
- To know how to use packaged applications in Helm charts

Niveau
Fondamental
Duration (in Days)
3
Reference
KUB-01