



Kubernetes Application Developer

Duration 3 day(s) (KUBERNETES)

Orchestrate containers with Kubernetes

Description

It is becoming more common to deploy applications in the form of containers. Kubernetes is an open source container orchestration tool designed to automate deploying, scaling, and operating containerized applications. The use of an orchestration tool is strongly recommended in order to better address the problem of load distribution and fault tolerance of applications deployed in the form of containers. This is to avoid reinventing the wheel and to benefit from the increase in productivity and responsiveness related to the use of containers. Kubernetes is one of the most cited products and used in this field. Its reputation and maturity are based on the experience and know-how of Google in terms of containers. The purpose of this training is to discover how to use it, understand the constraints associated with its use and how to develop the applications that will be deployed.

Goals

- Deploy simple applications (stateless and stateful)
- Deploy an application composed of several services
- Expose an application to the outside of the Kubernetes infrastructure
- Learn how to manage the data handled by the application in the Kubernetes infrastructure
- Update an application already deployed in Kubernetes
- Browse the best practices associated with the development of an application deployed in Kubernetes

Public

- Developers
- Tech Lead
- Technical Architects and Solutions
- Technical platform operators

Prerequisites

- Knowledge of containerization technologies, including Docker

Structure

50% Theory, 50% Practice

Program

Background

- Recalls on containers
- Containers without orchestration
- Orchestration features
- Orchestrators of the market
- Kubernetes
- Kubernetes Distributions
- Versions and APIs

Get started with Kubernetes

- Installation of local Kubernetes with mini kube
- Dashboard, CLI and API
- Start and expose a container

The pods

- Model / Concept of the pod Yaml and json descriptors
- Organization of pods with labels, selectors and namespaces
- Life cycle of pods
- Init Containers

ReplicaSets

- HealthChecks
- ReplicaSets
- DaemonSets
- Jobs

Services

- Expose the cluster internally
- Expose a service outside the Kubernetes instance (NodePort, LoadBalancer, Ingress)
- The concept of **readiness**

Volumes

- Simple data sharing between 2 containers of the same pod
- Access the fs of a cluster node
- Persistent Volumes and Persistent Volumes Claims

Configuration and secrets

- Environment variables
- ConfigMaps
- Secrets

Deployment strategies

- Provision of a new version of a pod
- Rolling imperative update
- Rolling Update declarative and object 'Deployment'

Stateful sets

Architecture and components

Required features in business

- Resource Management
- Horizontal self-scaling
- Compatible Applications Kubernetes
- Helm to simplify your deployments