



Kubernetes Application Developer

Duration 3 day(s) (KUBERNETES)

Orchestrate containers with Kubernetes

Description

It is becoming more common to deploy applications inside software 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 software 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 shell terminal and basic Linux commands
- Knowledge of containerization technologies, including Docker

Structure

50% Theory, 50% Practice

Program

Background

- Reminders on containers
- Containers without orchestration
- Orchestration capabilities
- Existing orchestrators
- Kubernetes
- Kubernetes Distributions
- API versioning

Get started with Kubernetes

- Local Kubernetes installation with mini kube
- Dashboard, CLI and API
- Start a container
- Expose the application

Pods

- The pod concept
- Descriptors
- Labels, Annotations, Namespaces
- Pod Lifecycle
- Init Containers

ReplicaSets

- HealthChecks
- ReplicationControllers vs ReplicaSets
- DaemonSets
- Jobs

Services

- Why services?
- Services for internal communication
- Expose service to the outside of the k8s instance
- Expose outside : Ingress vs LoadBalancer
- Readiness Probe
- Headless Services

Volumes

- Intro to Volumes
- Share data between containers of a pod
- Share data with host
- Persistent Volumes and Persistent Volumes Claims

Configuration and secrets

- Environment variables
- ConfigMaps
- Secrets

Deployment strategies

- Deployment strategies
- Rolling Update

Stateful sets

- Principles

Architecture and components

- Components

Enterprise ready features

- Managing hardware resources
- Auto-scaling

Develop compatible apps

- Kubernetes compatible apps
- Helm to simplify deployments
- Kustomize to manage deployments in different environments