



# Kubernetes Admin

Durée 3 jour(s) (KUBERNETES-ADM)

Gérer des plateformes Kubernetes

## Description

Déployer ses applications sous la forme de containers/conteneurs est de plus en plus courant, notamment dans les architectures techniques prônant l'organisation d'une application comme un ensemble de services collaborant entre eux. Afin de mieux répondre aux problématiques de répartition de charge et de tolérance à la panne des applications déployées sous forme de containers, l'utilisation d'un outil d'orchestration est fortement conseillée. L'utilisation d'une solution existante permet d'éviter de réinventer la roue et de pouvoir profiter de l'augmentation de productivité et de réactivité liées à l'utilisation des containers. Kubernetes est l'un des produits les plus cités et utilisés dans ce domaine, sa réputation et sa maturité s'appuient sur l'expérience et le savoir faire de Google en matière de containers. Cette formation couvre l'aspect opérations de l'installation et la maintenance d'une plateforme conteneurisée s'appuyant sur Kubernetes

## Objectifs

- Comprendre l'architecture et le fonctionnement interne d'une plateforme Kubernetes
- Savoir installer et opérer une instance Kubernetes
- Avoir toutes les clés pour bien choisir les différents outils liés à la maintenance d'une instance (réseau, logs, métriques)
- Savoir configurer et faire le suivi opérationnel d'une instance Kubernetes ()

## Public

- Opérateurs de plateforme technique

## Prérequis

- Connaissance des technologies de conteneurisation comme Docker
- Connaissance de Kubernetes, notamment des différents types de charges pouvant y être déployées (Pods, ReplicaSets, Deployments, Services, Ingress, DaemonSets, StatefulSets). Idéalement avoir suivi notre formation [Kubernetes Application Developer](#) ou une formation équivalente ou être certifié [Certified Kubernetes Application Developer](#).
- Bonne connaissance des systèmes linux ainsi que des connaissances de base concernant le réseau et les principes fondamentaux de sécurité sous linux

## Répartition

50% Théorie, 50% Pratique

## Programme

### Architecture

- Introduction
- Architecture
- Control Plane
- Workers

### Installation

- Installation steps
- Distributions
- Network solutions
- Container Runtimes
- DNS
- Storage

### Configuration and operational maintenance

- Logs centralization
- Metrics centralization
- Backup / Restore
- Tools for capacity planning (node resources consumption, namespaces, requests, limits)
- Garbage Collection
- Connectivity of a Kubernetes cluster with the rest of your infrastructure
- LoadBalancers
- Ingress
- Schedulers
- Troubleshooting the cluster

### Cluster Upgrade

- Kubernetes Release Cycle
- Upgrade process
- Upgrade with kubeadm
- Upgrade with kops
- Upgrade with kubespray
- Upgrade with gke
- Upgrade with eks

Day to day actions, make your Kubernetes users happy and

### aware

- Namespaces and Instances
- NetworkPolicies
- Common Admission Controllers
- LimitRanges
- Quotas
- Security contexts
- PodSecurityPolicy
- Authentication/Authorization/RBAC
- Persistent Volumes

### Extensibility: Operators, CRD and API Servers

- Controllers
- Dynamic Admission Controllers
- Custom Resource Definition
- Operators
- API Servers/Aggregation layer

### Federation, Service Mesh, Security