



Kubernetes Application Developer

Durée 3 jour(s) (KUBERNETES)

Orchestrer ses containers avec Kubernetes K8s

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 afin 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 a pour but de découvrir comment l'utiliser tout en comprenant quelles sont les contraintes associées à son utilisation, notamment sur la façon de développer les applications qui y seront déployées.

Formation animée en présentiel

La formation en présentiel se déroule sur des jours consécutifs

Formation disponible en mode "formation à distance"

La formation à distance peut se dérouler sur des jours consécutifs ou se décomposer en demies journées

Objectifs

- Définir les fonctionnalités d'un orchestrateur de conteneurs.
- Identifier l'architecture d'un cluster Kubernetes.
- Classifier et expérimenter chaque objet Kubernetes et leurs rôles respectifs.
- Construire une application composée de plusieurs services.

Public

- Développeurs
- Tech Lead
- Architectes techniques et solutions
- Opérateurs de plateforme technique

Prérequis

- Connaissances du shell et manipulations de base Linux
- Connaissances des technologies de conteneurisation, notamment Docker

Répartition

50% Théorie, 50% Pratique

Evaluations des acquis

L'évaluation des acquis de la formation se fera en séance au travers d'ateliers, d'exercices et/ou de travaux pratiques. Dans le cas d'une formation officielle éditeur, veuillez nous consulter afin que nous vous fassions part des modalités d'évaluation.

A l'issue de la formation, vous sera transmis une évaluation à chaud de l'action de formation qui vous permettra de nous faire part de vos retours quant à votre expérience apprenant avec Zenika.

Ressources pédagogiques

Les ressources pédagogiques proviennent de productions des équipes Zenika et/ou de la documentation éditeur dans le cas d'une formation "Officielle". Les documents sont en français ou en anglais.

RQTH et ma formation Zenika

Si vous êtes sujet à un handicap, prenez contact avec nos équipes pour que nous puissions définir ensemble comment nous pourrions aménager la session afin que vous puissiez vivre une expérience en formation inchangée.

Programme

Contexte

- Rappels sur les containers
- Containers sans orchestration
- Fonctionnalités d'orchestration
- Orchestrateurs du marché
- Kubernetes
- Distributions Kubernetes
- Versions et API

Architecture et composants

Premiers pas avec Kubernetes

- Installation de Kubernetes en local avec minikube
- Dashboard, CLI et API
- Démarrer et exposer un container

Les Pods

- Modèle/Concept du Pod
- Descripteurs YAML et JSON
- Organisation des Pods avec les labels, les sélecteurs et les Namespaces
- Security Context
- Init Containers
- Cycle de vie des Pods
- HealthChecks

Droits et comptes de service

- Descripteurs liés au RBAC
- Comptes de service

#Controllers

- ReplicaSets
- DaemonSets
- Jobs

Services

- Exposer en interne du cluster
- Exposer un service vers l'extérieur de l'instance Kubernetes (NodePort, LoadBalancer, Ingress)
- Sécuriser le trafic au sein du cluster avec les NetworkPolicies
- Le concept du **readiness**

Volumes

- Partage simple de données entre 2 containers d'un même Pod
- Accéder au fs d'un nœud du cluster
- Persistent Volumes et Persistent Volumes Claims

Configuration et secrets

- Variables d'environnements
- ConfigMaps
- Secrets

Stratégies de déploiement

- Mise à disposition d'une nouvelle version d'un Pod
- Rolling Update impératif
- Rolling Update déclaratif et 'Deployment'

Stateful sets

Limites & Autoscaler

- Gestion des ressources
- Auto-scaling horizontal

Helm & Kustomize

- Helm pour simplifier vos déploiements
- Kustomize pour gérer vos déploiements dans différents environnements

Conclusion

- Résumé
- Bonnes pratiques
- Kubernetes, et après ?