Orchestrateur de conteneurs : les fondamentaux de Kubernetes

Prix : 1 450 €HT

Durée : 2 jours

Code de Référence : KUBFON

Catalogue Cloud Native et DevOps

Les conteneurs et Kubernetes sont devenus des fondations majeures des architectures applicatives modernes. Kubernetes permet d'orchestrer le déploiement, la mise à l'échelle et l'exploitation d'applications conteneurisées, tout en apportant des mécanismes de standardisation, de résilience et de portabilité entre environnements.

Cette formation de deux jours donne aux participants les bases nécessaires pour comprendre le rôle de Kubernetes, manipuler ses principaux objets et appréhender les usages essentiels d'un orchestrateur de conteneurs dans un contexte Cloud Native. Elle alterne apports théoriques, démonstrations et exercices guidés afin de faciliter la prise en main des notions fondamentales, des commandes usuelles et des premières bonnes pratiques d'exploitation.

Objectifs de la formation

A l'issue de cette formation, vous serez capable de :

- Décrire le rôle de Kubernetes dans l'orchestration d'applications conteneurisées.
- Identifier les composants principaux d'un cluster Kubernetes et leur fonction.
- Utiliser les commandes essentielles de kubectl pour interagir avec un cluster.
- Créer et manipuler les objets fondamentaux Kubernetes : pods, namespaces, deployments et services.
- Expliquer les principes de base de l'exposition applicative avec les services.
- Gérer les configurations, secrets et volumes dans un contexte Kubernetes simple.

- Appliquer les premières bonnes pratiques d'organisation, de sécurité et d'exploitation d'un environnement Kubernetes.

- Situer les notions à approfondir pour aller vers des usages avancés : observabilité, CI/CD, GitOps, sécurité et plateformes managées.

Public

Cette formation s'adresse aux :

- Administrateurs systèmes souhaitant découvrir les fondamentaux de Kubernetes,
- Développeurs amenés à déployer ou maintenir des applications conteneurisées,
- Ingénieurs systèmes et réseaux impliqués dans des projets Cloud Native ou DevOps,
- Architectes techniques souhaitant disposer d'une vision synthétique de Kubernetes,
- Chefs de projets IT souhaitant mieux comprendre les apports et limites de Kubernetes,
- Consultants Cloud ou DevOps débutant sur les technologies d'orchestration de conteneurs.

Prérequis

Avoir connaissance de la terminologie et des concepts d'architecture informatique.
Avoir des connaissances basiques des environnements Linux.
Connaître les principes généraux des conteneurs et de Docker.
Être à l'aise avec l'utilisation d'un terminal ou d'une ligne de commande.
Une première expérience des environnements Cloud ou DevOps est un plus.

Programme de la formation

Les démonstrations, exemples et travaux pratiques sont donnés à titre indicatif et pourront être adaptés selon les besoins et le contexte de la session.

Jour 1
Matinée – Fondamentaux de Kubernetes et architecture d'un cluster
Introduction – Positionnement et cadrage de la formation

- Présentation des objectifs et du déroulé des deux journées
- Tour de table des participants
- Rappel des notions utiles : conteneur, image, registre, réseau, terminal, Cloud
- Présentation du rôle de Kubernetes dans les architectures Cloud Native et les pratiques DevOps

Chapitre 1 – Comprendre le rôle de Kubernetes

- Origine et positionnement de Kubernetes dans l'écosystème des conteneurs
- Limites d'une gestion manuelle des conteneurs à grande échelle
- Objectifs d'un orchestrateur : déploiement, supervision, redémarrage, mise à l'échelle, exposition et résilience
- Cas d'usage adaptés à Kubernetes : applications conteneurisées, microservices, plateformes Cloud Native, environnements standardisés

- Cas d’usage moins adaptés : applications simples, contraintes fortes de persistance, charges isolées ou environnements peu industrialisés
- Panorama de l’écosystème Kubernetes : CNCF, distributions, plateformes managées et outils complémentaires

Démonstrations, exemples et travaux pratiques :

- Illustration du passage d’un conteneur isolé à une application orchestrée
- Présentation d’un cluster Kubernetes et de ses principaux objets
- Lecture commentée d’un premier manifeste Kubernetes simple

Chapitre 2 – Découvrir l’architecture Kubernetes et manipuler kubectl

- Architecture générale d’un cluster Kubernetes
- Rôle du control plane et des worker nodes
- Composants principaux : API Server, scheduler, controller manager, etcd, kubelet, kube-proxy
- Présentation de kubectl et du fichier de configuration kubeconfig
- Commandes essentielles : consulter le cluster, lister les objets, décrire une ressource, consulter les logs
- Notion de namespace pour organiser les ressources

Démonstrations, exemples et travaux pratiques :

- Vérification de l’accès au cluster Kubernetes
- Utilisation des commandes kubectl get, kubectl describe et kubectl logs
- Création et consultation d’un namespace
- Observation de l’état des ressources du cluster

Après-midi – Objets fondamentaux et premiers déploiements
Chapitre 3 – Déployer une application avec les objets Kubernetes fondamentaux

- Comprendre la notion de pod
- Créer et exécuter un pod simple
- Comprendre le rôle d’un deployment
- Déployer une application via un deployment
- Mettre à jour une application et observer le mécanisme de rollout
- Revenir à une version précédente avec le rollback
- Comprendre le rôle des labels et selectors
- Identifier les erreurs fréquentes : image inaccessible, mauvais port, mauvaise indentation YAML, ressources mal nommées

Démonstrations, exemples et travaux pratiques :

- Lecture commentée d’un manifeste de pod et de deployment
- Déploiement d’une application conteneurisée simple
- Mise à jour d’une image applicative et observation du rollout
- Correction d’erreurs courantes dans un manifeste Kubernetes
- Modification du nombre de réplicas
- Suppression des ressources créées

Chapitre 4 – Exposer une application avec les services Kubernetes

- Comprendre le rôle des services Kubernetes
- Différencier ClusterIP, NodePort et LoadBalancer
- Comprendre la résolution de service et la communication entre pods
- Introduction à l'exposition HTTP
- Premiers principes de routage applicatif dans Kubernetes

Démonstrations, exemples et travaux pratiques :

- Création d'un service pour exposer une application
- Vérification de l'accès à l'application déployée
- Comparaison entre ClusterIP, NodePort et LoadBalancer
- Analyse d'un schéma d'exposition applicative

Jour 2
Matinée – Configuration, secrets, volumes et exploitation de base
Chapitre 5 – Gérer la configuration et les secrets

- Gérer la configuration applicative avec ConfigMaps
- Gérer les informations sensibles avec Secrets
- Injecter des variables d'environnement dans un pod
- Monter une configuration sous forme de fichier
- Identifier les limites des Secrets Kubernetes natifs
- Bonnes pratiques de base : séparation configuration / code, gestion des secrets, organisation des manifestes

Démonstrations, exemples et travaux pratiques :

- Ajout d'une ConfigMap dans un manifeste applicatif
- Utilisation d'un Secret dans une application
- Modification d'une configuration et observation de son impact
- Identification collective des bonnes pratiques à appliquer sur un ensemble de manifestes fourni

Chapitre 6 – Comprendre la persistance et les volumes

- Comprendre les principes de base de la persistance dans Kubernetes
- Différence entre volume, PersistentVolume et PersistentVolumeClaim
- Cycle de vie d'un volume et d'un conteneur
- Cas d'usage des volumes : configuration, données applicatives, stockage temporaire ou persistant
- Points de vigilance : suppression des ressources, sauvegarde, performance, dépendance à l'infrastructure
- Introduction aux Storage Classes

Démonstrations, exemples et travaux pratiques :

- Ajout d'un volume à une application simple
- Création ou lecture d'un PersistentVolumeClaim
- Vérification de la persistance des données
- Analyse des risques liés à la suppression d'un pod ou d'un volume
- Lecture commentée d'un manifeste utilisant une StorageClass

Après-midi – Bonnes pratiques, exploitation et ouverture vers les usages avancés
Chapitre 7 – Premières bonnes pratiques d'exploitation Kubernetes

- Organisation des manifestes Kubernetes
- Structuration par namespace, application ou environnement
- Bonnes pratiques de nommage, labels et annotations
- Limitation des privilèges et principes de sécurité de base
- Nettoyage des ressources et gestion du cycle de vie applicatif
- Introduction aux ressources requests et limits
- Premiers réflexes de diagnostic : logs, events, describe, état des pods

Démonstrations, exemples et travaux pratiques :

- Analyse d'un ensemble de manifestes Kubernetes
- Identification des améliorations possibles : organisation, lisibilité, sécurité, maintenabilité
- Utilisation de kubectl describe, kubectl logs et des events pour diagnostiquer un problème simple
- Correction guidée d'un déploiement présentant une erreur courante

Chapitre 8 – Mise en perspective : plateformes, outillage et trajectoire Kubernetes

- Présentation des plateformes Kubernetes managées : AKS, EKS, GKE
- Place de Kubernetes dans une plateforme CaaS ou Cloud Native
- Introduction à Helm pour le packaging applicatif
- Introduction à l'observabilité : logs, métriques, alerting
- Ouverture vers CI/CD et GitOps
- Introduction aux enjeux de sécurité Kubernetes
- Limites d'une approche manuelle et besoins d'industrialisation

Démonstrations, exemples et travaux pratiques :

- Lecture commentée d'un chart Helm ou d'une structure applicative industrialisée
- Présentation d'un pipeline ou d'un flux GitOps simplifié
- Analyse d'une trajectoire de montée en maturité Kubernetes
- Élaboration collective d'une liste de prochaines étapes pour intégrer Kubernetes dans un SI

Conclusion – Synthèse et plan d'action individuel

- Synthèse des notions clés des deux journées
- Présentation des pistes d'approfondissement : Helm, sécurité Kubernetes, observabilité, CI/CD, GitOps, Kubernetes managé, plateformes CaaS
- Questions / réponses
- Auto-évaluation finale

Méthodes pédagogiques

- Accompagnement théorique et pratique avec une pédagogie alliant théorie, démonstrations et exercices guidés, puis mise en pratique des notions abordées avec une évaluation des manipulations réalisées.

- Le formateur tient compte de la situation de chaque apprenant et se base sur les expériences, les connaissances et les questions particulières des participants pour nourrir le groupe de cas concrets et de retours d'expériences ciblées.

- Le format de deux jours permet une prise en main progressive des fondamentaux Kubernetes : architecture du cluster, commandes usuelles, objets principaux, exposition applicative, configuration, persistance et premières bonnes pratiques d'exploitation.

Les supports de formation seront fournis aux participants au cours de la formation au format PDF.

Méthodes d'évaluation des acquis

Avant la formation :

- Recueil éventuel des attentes et du contexte des participants en amont de la session.

En cours de formation :

- Points d'étapes réguliers par le formateur sur la compréhension des stagiaires.
- Vérification des résultats obtenus lors des exercices guidés.
- Retour d'expérience en fin de journée et lors de la conclusion de la formation.

Après la formation « à chaud » :

- Questionnaire d'auto-évaluation des compétences complété individuellement par chaque stagiaire après la formation.
- Questionnaire de satisfaction « à chaud » complété individuellement par chaque stagiaire en fin de formation.
- Le compte rendu formateur complété par le formateur.

Après la formation « à froid » :

- Questionnaire de satisfaction « à froid » complété individuellement par chaque stagiaire quelques semaines après la session de formation.

Un certificat de réalisation de fin de formation est remis au stagiaire lui permettant de faire valoir le suivi de la formation.