



# kubernetes

## EN BREF

**Référence** : K8S

**Durée** : 3 jours

**Niveau** : expert

**Lieu** : Paris, Orléans, sur site ou à distance

**Dates** : voir site internet ou nous contacter

**Tarif** (inter-entreprises) : 2100 € HT

**Mise à jour** : 04/11/2025

## NOS FORMATEUR-RICE-S



Expert·e·s dans leur domaine, ils et elles ont l'expérience du développement logiciel et de la réalisation de projets informatiques.



Ils et elles ont dispensé, avec succès, de nombreuses heures de formation vis-à-vis d'un public exigeant de : d'enseignant·e·s, de technicien·ne·s, d'ingénieur·e·s, de scientifiques, développeur·euse·s, administrateur·trice·s IT.

## INSCRIPTION / RENSEIGNEMENTS



**02 38 21 30 57**  
(du lundi au vendredi)



**contact@numgrade.com**



## MAÎTRISEZ L'ORCHESTRATION DES CONTENEURS AVEC KUBERNETES !

Ce stage intensif vous permettra de déployer des applications de manière fluide et scalable.

## LE CONTEXTE

Avec la croissance de l'usage du numérique et l'augmentation des besoins de gestion de données, les équipes informatiques font face à de nouveaux défis :

- Une multiplication des applications métiers critiques,
- Une exposition accrue aux risques de sécurité,
- Une attente forte autour de la résilience et de la rapidité de déploiement,
- Une attente de haute disponibilité et de scalabilité.

Il devient alors impératif de maintenir le système opérationnel, d'assurer des déploiements rapides, et de garantir un haut niveau de sécurité. L'Infrastructure as Code (IaC) et la conteneurisation répondent à ces besoins. Pour les équipes utilisant Docker, **Kubernetes (k8s) est une solution naturelle pour automatiser la gestion, le déploiement et la montée en charge des applications.**

## LA FORMATION

Cette formation vous permettra de **déployer des applications, suivre les logs, versionner ; automatiser la mise en place et la mise à jour, gérer automatiquement les certificats ; gérer les secrets de manière sécurisée ; mettre en œuvre une chaîne de déploiement continue.**

Elle est extensible, et ouvre la voie vers la gestion des volumes, des droits, des politiques réseau, etc.

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation, le·la stagiaire sera capable de :

- Comprendre les problématiques que Kubernetes permet de résoudre.
- Maîtriser les objets fondamentaux : Pod, ReplicaSet, Deployment, Service, Ingress.
- Déployer une application de bout en bout (de docker-compose à k8s).
- Utiliser les outils Helm et Kustomize pour gérer des déploiements.





# kubernetes

## NOS PRESTATIONS

- ✓ Formations en INTER ou INTRA
- ✓ Développement logiciel
- ✓ Conseil et assistance technique

## ÉVALUATION DE LA FORMATION

À l'issue de chaque demi-journée de formation, une **feuille d'émargement** est signée, à la fois par le·la stagiaire et par le·la formateur·rice.

Des **QCMs** en début et fin de formation permettent de vérifier l'**acquisition des compétences** durant la formation.

Un **certificat de formation**, ainsi qu'une **attestation de compétences** sont remis aux stagiaires en fin de formation, leur permettant de **faire valoir l'acquisition de leurs nouvelles connaissances**.

## EXIGEZ LA DIFFÉRENCE NUMGRADE

- ✓ Des consultant·e·s expert·e·s et passionné·e·s
- ✓ Concentré·e·s sur la réussite de leurs clients et clientes
- ✓ En quête permanente d'excellence technique

## SITUATION DE HANDICAP

NUMGRADE s'engage à accueillir les personnes en situation de handicap. N'hésitez pas à nous contacter pour que le suivi de la formation se déroule dans les meilleures conditions possibles.

- Connaître une architecture Kubernetes concrète et fonctionnelle pour pouvoir collectivement décider des directions de l'équipe.

## PUBLIC CONCERNÉ

Développeurs, ingénieurs DevOps, administrateurs systèmes souhaitant monter en compétence sur l'orchestration de conteneurs, chefs de projet techniques ou architectes souhaitant comprendre les enjeux liés à Kubernetes.

## PRÉREQUIS

- Avoir des connaissances de base en ligne de commande Linux, ou avoir suivi la formation « Linux débutant ».
- Avoir une compréhension générale de Docker (build d'image, docker-compose, volumes, réseaux), ou avoir suivi la formation « Docker débutant ».
- Avoir des notions de réseau, d'API REST et de gestion de configuration.

Un questionnaire ou un entretien de pré-formation est proposé pour valider l'adéquation entre le besoin et la formation.

## PRÉREQUIS TECHNIQUES

Les logiciels suivants devront être installés sur votre machine : helm, docker, minikube, kubectl, ainsi qu'un éditeur de code. Vous devez être muni d'un accès internet permettant à minima l'accès à des sites tels que dockerhub, gitlab, github, etc.

## PROGRAMME DE LA FORMATION :

### Introduction et fondamentaux

- Pourquoi Kubernetes ? Retour sur les limites de Docker seul
- Architecture de Kubernetes (Control Plane, Nodes, etcd, API Server, Scheduler, Controller Manager).
- Principe des Controllers
- Namespaces et objets de base : Pod, ReplicaSet, Deployment, Service, Ingress
- Installation locale (minikube, docker desktop).
- Installation d'ArgoCD
- Expérimentation : création de premiers objets, utilisation de kubectl





# kubernetes

## MOYENS PÉDAGOGIQUES

Cette formation s'appuie sur une pédagogie hybride :

- **Apports théoriques ciblés** pour construire des bases solides.
- **Exercices pratiques guidés** pour expérimenter directement les concepts.
- **Ateliers collectifs** pour favoriser la prise de recul, la réflexion partagée et l'ancrage dans un contexte d'équipe.

**Une architecture minimaliste mais réaliste sert de fil conducteur** : Un cluster Kubernetes avec ArgoCD, nginx, cert-manager, external-secrets, des applications décrites en Helm, et GitLab comme socle DevOps.

Cette base permet de :

- **Déployer des applications, suivre les logs, versionner.**
- **Automatiser la mise en place et la mise à jour, gérer automatiquement les certificats.**
- **Gérer les secrets de manière sécurisée.**
- **Mettre en œuvre une chaîne de déploiement continue.**

Elle est extensible, et ouvre la voie vers la gestion des volumes, des droits, des politiques réseau, etc.

## FORMATIONS CONNEXES

✓ [Linux débutant](#)

✓ [Docker Débutant](#)

✓ [Ansible](#)

## Première application et Helm

- Création d'un Helm chart basique
- Déploiement via ArgoCD
- Analyse des composants : Deployment, Service, Ingress
- Expérimentation : mises à jour, débogage
- Passage d'un docker-compose.yml à un manifest Kubernetes

## Gérer la persistance des données

- Éviter la persistance locale autant que possible
  - Pourquoi ?
  - Configmap & secrets
  - Utilisation d'une base de données extérieure
- Quand on ne peut pas éviter la persistance : les volumes
  - Objets Kubernetes pour la persistance (Persistent volumes)

## Gestion des secrets

- Problématiques liées à la gestion des secrets :
  - Secrets applicatifs (env vars, fichiers, montages)
  - Accès aux registres privés (tokens, credentials)
- Déploiement de Secrets Kubernetes classiques "à la main"
- Introduction aux solutions avancées :
  - Sealed Secrets (Bitnami)
  - External Secrets Operator (ESO)
- Mise en pratique

## Organisation en équipe

- Travail collectif pour la détermination d'un plan d'action :
  1. Quels objectifs à court et moyen terme ?
  2. Choix d'une architecture initiale pragmatique convenant à l'équipe reposant sur GitOps et ArgoCD :
    - Gestion des rôles et accès utilisateurs
    - Structure des dépôts Git (app of apps, mono-repo, multi-repo)
    - Structure générique des helms charts
    - Nommage des objets, des projets, etc.

