

---

## Programme de Formation

---

# Kubernetes : déploiement et orchestration d'applications



---

## Organisation

**Durée :** 21 heures

**Mode d'organisation :** Présentiel

---

## Contenu pédagogique



### **Public visé**

Ingénieurs développement et déploiement, administrateurs systèmes, architectes logiciels, responsables infrastructure IT, chefs de projets techniques souhaitant maîtriser l'orchestration de conteneurs en production. Formation adaptée aux professionnels impliqués dans la transformation digitale et la migration vers le cloud.



### **Objectifs pédagogiques**

- Concevoir une architecture micro-services adaptée aux environnements cloud-native
- Déployer et orchestrer des applications conteneurisées avec Kubernetes
- Configurer la persistance de données et les services réseau dans un cluster Kubernetes
- Mettre en œuvre des stratégies de déploiement avancées (rolling update, blue/green, canary)
- Analyser et surveiller les performances d'applications avec des outils d'observabilité



### **Description**

#### **1er jour : Fondamentaux et Architecture (7h)**

##### Introduction aux micro-services (1h30)

- Théorie (45min) :
  - Architecture monolithique vs micro-services : avantages et inconvénients
  - Cas d'usage et critères de choix
  - Les 12 facteurs pour les applications cloud-native
- Atelier pratique (45min) :
  - Analyse d'une architecture existante
  - Identification des composants à découper
  - Stratégie de migration progressive

##### Conteneurs et écosystème (2h)

- Rappels Docker (1h) :



- Vue d'ensemble : Docker vs containerd vs autres runtimes
- Cycle de vie des conteneurs
- Bonnes pratiques de création d'images
- Travaux pratiques (1h) :
  - Installation et configuration Docker
  - Création d'images optimisées (multi-stage builds)
  - Gestion des registres privés/publics
  - Sécurisation des conteneurs

#### Kubernetes - Architecture et concepts de base (3h)

- Architecture Kubernetes (1h) :
  - Control Plane : API Server, etcd, Scheduler, Controller Manager
  - Nodes : kubelet, kube-proxy, container runtime
  - Réseaux et stockage dans Kubernetes
- Installation et premiers pas (1h) :
  - Options d'installation : minikube, kind, k3s, clusters managés
  - Configuration kubectl et contexts
  - Cycle de vie d'une requête kubectl
- Déploiement d'applications (1h) :
  - Création de Pods
  - Exposition réseau avec Services
  - Passage à l'échelle horizontal
  - TP : Déployer votre première application web

#### Exploration et debugging (30min)

- Commandes de diagnostic essentielles
- Logs et événements
- TP : Résoudre des problèmes courants

### **2e jour : Concepts avancés et déploiement (7h)**

#### Objets Kubernetes avancés 2h30

- Gestion des Pods (45min) :
  - Cycle de vie et états
  - Init containers et sidecars
  - Sondes de santé (liveness, readiness, startup)
- Configuration et sécurité (45min) :
  - ConfigMaps : externaliser la configuration
  - Secrets : gestion sécurisée des données sensibles
  - Variables d'environnement et volumes
- Services réseau (60min) :
  - Types de services : ClusterIP, NodePort, LoadBalancer
  - Services Headless pour bases de données
  - Ingress Controllers et routage HTTP/S
  - TP : Configuration complète d'une application multi-tiers

#### Organisation et gouvernance (1h)

- Labels et sélecteurs :
  - Stratégies de nommage
  - Filtrage et requêtes
- Namespaces :

- Isolation logique
- Quotas et limites de ressources
- TP : Organisation d'environnements (dev/staging/prod)

#### Stratégies de déploiement (3h)

- Contrôleurs de déploiement (1h) :
  - ReplicaSets : garantir la disponibilité
  - DaemonSets : services système
  - Jobs et CronJobs : tâches batch et calcul intensif
- Déploiements avancés (1h30) :
  - Deployments et rolling updates
  - Stratégies blue/green et canary
  - Rollback automatique
  - TP : Mise en œuvre rolling update avec surveillance
- GitOps et déploiement continu (30min) :
  - Principes GitOps
  - Démo ArgoCD : Déploiement automatisé depuis Git

#### Atelier intégré (30min)

- Application multi-composants avec base de données
- Configuration réseau complète
- Stratégie de mise à jour

### **3e jour : Production et observabilité (7h)**

#### Persistance et données 2h30

- Stockage Kubernetes (1h) :
  - Volumes éphémères vs persistants
  - PersistentVolumes et PersistentVolumeClaims
  - StorageClasses et provisioning dynamique
- Bases de données distribuées (1h30) :
  - StatefulSets : ordre et identité
  - Patterns pour la haute disponibilité
  - TP pratique : Déploiement MongoDB cluster avec réplication

#### Observabilité complète (4h)

- Fondamentaux de l'observabilité (45min) :
  - Les trois piliers : logs, metrics, traces
  - OpenTelemetry : standard et bonnes pratiques
  - Stratégies d'instrumentation
- Monitoring et métriques (1h15) :
  - Architecture Prometheus :
    - Modèle de données et PromQL
    - Service discovery dans Kubernetes
    - Alerting avec AlertManager
  - Démo : Installation et configuration Prometheus
- Gestion des logs (1h) :
  - Agrégation avec FluentD :
    - Collecte depuis les conteneurs
    - Parsing et enrichissement

- Routing vers différents backends
  - Démo : Pipeline de logs centralisé
- Tracing distribué (30min) :
  - OpenTelemetry Operator :
    - Auto-instrumentation
    - Corrélation des traces
    - Démo : Traçage d'une requête micro-services
- Visualisation unifiée (30min) :
  - Grafana avancé :
    - Dashboards pour Kubernetes
    - Alerting et notifications
    - Corrélation multi-sources
  - Démo : Dashboard de production complet

#### Administration et troubleshooting (30min)

- Monitoring du cluster :
  - Santé des composants (API server, etcd, nodes)
  - Métriques de performance et capacité
  - TP : Identifier et résoudre des pannes courantes
- Bonnes pratiques opérationnelles :
  - Sauvegardes et disaster recovery
  - Mise à jour des clusters
  - Sécurité en production



### **Prérequis**

Connaissances de base en administration système Linux, notions de virtualisation, familiarité avec les concepts réseau (TCP/IP, DNS). Expérience recommandée avec les conteneurs Docker ou équivalent. Pratique de la ligne de commande indispensable. Avoir déjà développé ou déployé des applications web constitue un plus.



### **Modalités pédagogiques**

- Répartition : 8h cours magistraux, 13h de labs et de pratique
- Organisation des Labs : groupes de 15 participants maximum, 1 intervenant
- Présentiel OU à distance (choix selon nombre d'inscrits et préférences)



### **Moyens et supports pédagogiques**

- Supports de cours PDF, scripts et manifests YAML, guide de référence Kubernetes, accès plateforme de TP en ligne, liens vers documentation officielle.
- Equipements/logiciels mis à disposition : accès cluster Kubernetes partagé, machines virtuelles préconfigurées, outils monitoring (Prometheus, Grafana).
- Équipements que les participants devront amener : ordinateur portable (Linux/macOS/Windows), installation d'un client SSH, connexion internet stable.



### **Modalités d'évaluation et de suivi**

Un suivi individualisé par des évaluations formatives est assuré. Une attestation de fin de formation est délivrée à la fin du parcours.



### **Informations sur l'admission**

L'admission à cette formation ne fait l'objet d'aucun examen, test ou sélection préalable ; l'inscription est validée après réception du dossier complet et confirmation par l'organisme de formation.



### ***Informations sur l'accessibilité***

Notre organisme s'engage à garantir l'accessibilité de ses formations à distance et en présentiel aux personnes en situation de handicap. Un référent handicap est mobilisable afin d'analyser les besoins spécifiques et de mettre en place, lorsque cela est possible, les adaptations pédagogiques, techniques ou organisationnelles nécessaires.