TELEMACH LEARNING

Orchestration de conteneurs avec Kubernetes

Programme de formation avancée — **DEVOPS-010** — 3 jours / 21 heures

Architecture K8s

Comprendre composants et design

Déploiement

Gérer pods, services et CI/CD

Observabilité

Logs, metrics et traçage en prod

INFORMATIONS GÉNÉRALES

Fiche de formation

Code	DEVOPS-010
Durée	21 heures — 3 jours
Format	Intra-entreprise — Présentiel ou distanciel
Public cible	Ingénieurs DevOps, cloud engineers, tech leads
Prérequis	Maîtrise Docker (DEVOPS-001 recommandé)
Niveau	Avancé
Financement	OPCO — Plan de Développement des Compétences
Formateur	Alexandre BOIGUES
Organisme	Telemach Learning — NDA 11 92 27843 92

À qui s'adresse cette formation ?

Cette formation s'adresse aux professionnels techniques souhaitant maîtriser Kubernetes en environnement de production.

- Ingénieurs DevOps en activité
- Cloud engineers
- Tech leads et architectes

ⓘ Prérequis : maîtrise de Docker indispensable. La formation DEVOPS-001 est fortement recommandée.

OBJECTIFS PÉDAGOGIQUES

Ce que vous saurez faire à l'issue de la formation

À l'issue de ces 3 jours de formation intensive, chaque participant sera capable de concevoir, déployer et opérer des applications conteneurisées sur Kubernetes en environnement professionnel.

1	**Architecture Kubernetes** Comprendre le rôle de chaque composant : etcd, API server, scheduler, kubelet et controller manager.
2	**Déploiement d'applications** Utiliser les ressources adaptées : Deployment, StatefulSet, DaemonSet, Jobs et CronJobs.
3	**Exposition des services** Maîtriser ClusterIP, NodePort, LoadBalancer et la Kubernetes Gateway API.
4	**Configuration & Volumes** Gérer ConfigMaps, Secrets et volumes persistants (PV / PVC / StorageClass).
5	**Helm** Packager et déployer des applications avec Helm : charts, templates, values et releases.
6	**Observabilité** Mettre en place Prometheus, Grafana et la centralisation des logs pour surveiller le cluster.

MODULE 1 — 3H

Architecture Kubernetes

Control Plane

API Server

Point d'entrée unique pour toutes les interactions avec le cluster. Expose l'API REST Kubernetes.

etcd

Base de données clé-valeur distribuée stockant l'état complet du cluster.

Scheduler

Assigne les pods aux nœuds disponibles selon les contraintes de ressources.

Controller Manager

Boucle de contrôle maintenant l'état désiré du cluster.

Nœuds Workers

kubelet

Agent s'exécutant sur chaque nœud, responsable du cycle de vie des pods.

kube-proxy

Gère les règles réseau pour permettre la communication entre services.

Container Runtime

Moteur d'exécution des conteneurs (containerd, CRI-O).

Atelier pratique : création d'un cluster local avec **kind** ou **minikube** et prise en main de `kubectl` (get, describe, apply, delete, logs).

MODULE 2 — 5H

Workloads : Pods, Deployments, StatefulSets

Le module le plus dense de la formation couvre l'ensemble des ressources de charge de travail Kubernetes, des plus simples aux plus avancées.

Pods

Unité de base Kubernetes. Multi-conteneurs, init containers, probes de liveness et readiness.

Deployments

Rolling update, rollback automatique, HPA pour l'auto-scaling horizontal selon la charge.

StatefulSets

Applications avec état persistant : bases de données, brokers de messages, stockage distribué.

DaemonSets

Déploiement d'un pod sur chaque nœud du cluster : agents de monitoring, collecteurs de logs.

Jobs & CronJobs

Tâches ponctuelles ou planifiées : migrations, sauvegardes, traitements batch.

> **Atelier fil rouge :** Déploiement d'une API REST avec haute disponibilité, rolling update et configuration du HPA.

MODULE 3 — 4H

Services et API Gateway

Types de services Kubernetes

ClusterIP

Exposition interne uniquement. Idéal pour la communication inter-services.

NodePort

Exposition sur un port statique de chaque nœud. Utile pour les tests et environnements on-premise.

LoadBalancer

Intégration avec le load balancer du cloud provider pour une exposition publique.

Kubernetes Gateway API

La nouvelle génération de routage HTTP dans Kubernetes, plus expressive et extensible que l'Ingress classique.

GatewayClass

Définit le type de gateway et le contrôleur associé.

Gateway

Instance de point d'entrée réseau avec configuration TLS.

HTTPRoute

Règles de routage HTTP vers les services backend.

Les **Network Policies** permettent de segmenter le réseau et de contrôler les flux entre namespaces et pods.

MODULE 4 — 4H

Configuration, Secrets et Volumes

La gestion de la configuration et du stockage est un pilier fondamental pour opérer des applications en production sur Kubernetes.

ConfigMaps

Externaliser la configuration des applications hors des images Docker.

- Variables d'environnement
- Fichiers de configuration montés
- Mise à jour sans redéploiement

Secrets

Gérer les données sensibles de manière sécurisée.

- Credentials et mots de passe
- Certificats TLS
- Tokens d'API

Volumes Persistants

Stockage découplé du cycle de vie des pods.

- PersistentVolume (PV)
- PersistentVolumeClaim (PVC)
- StorageClass dynamique

RBAC & Namespaces

Contrôle des accès et isolation des environnements.

- Namespaces par équipe/env
- Roles et ClusterRoles
- ServiceAccounts

MODULE 5 — 3H

Helm : gestion des packages Kubernetes

Qu'est-ce que Helm ?

Helm est le gestionnaire de packages officiel de Kubernetes. Il permet de packager, distribuer et gérer des applications complexes sous forme de **charts**.

> Helm, c'est à Kubernetes ce qu'apt ou npm est à Linux et Node.js.

Helm simplifie drastiquement le déploiement d'applications tierces comme PostgreSQL, Redis ou des API Gateways.

Cycle de vie avec Helm

1

Créer / Récupérer un chart

Helm Hub, Artifact Hub ou `helm create` pour un chart personnalisé.

2

Installer une release

`helm install` avec un fichier `values.yaml` personnalisé.

3

Mettre à jour

`helm upgrade` pour déployer une nouvelle version sans interruption.

4

Rollback & Uninstall

`helm rollback` en cas de problème, `helm uninstall` pour nettoyer.

MODULE 6 — 2H

Monitoring, Logging et Observabilité

Un cluster Kubernetes en production sans observabilité est un cluster aveugle. Ce module couvre les outils essentiels pour surveiller, analyser et déboguer vos applications.

Prometheus

Collecte de métriques de cluster et d'application. Alerting basé sur des règles PromQL. Intégration native avec Kubernetes.

Grafana

Dashboards visuels pour les métriques Prometheus. Tableaux de bord préconfigurés pour Kubernetes (nœuds, pods, namespaces).

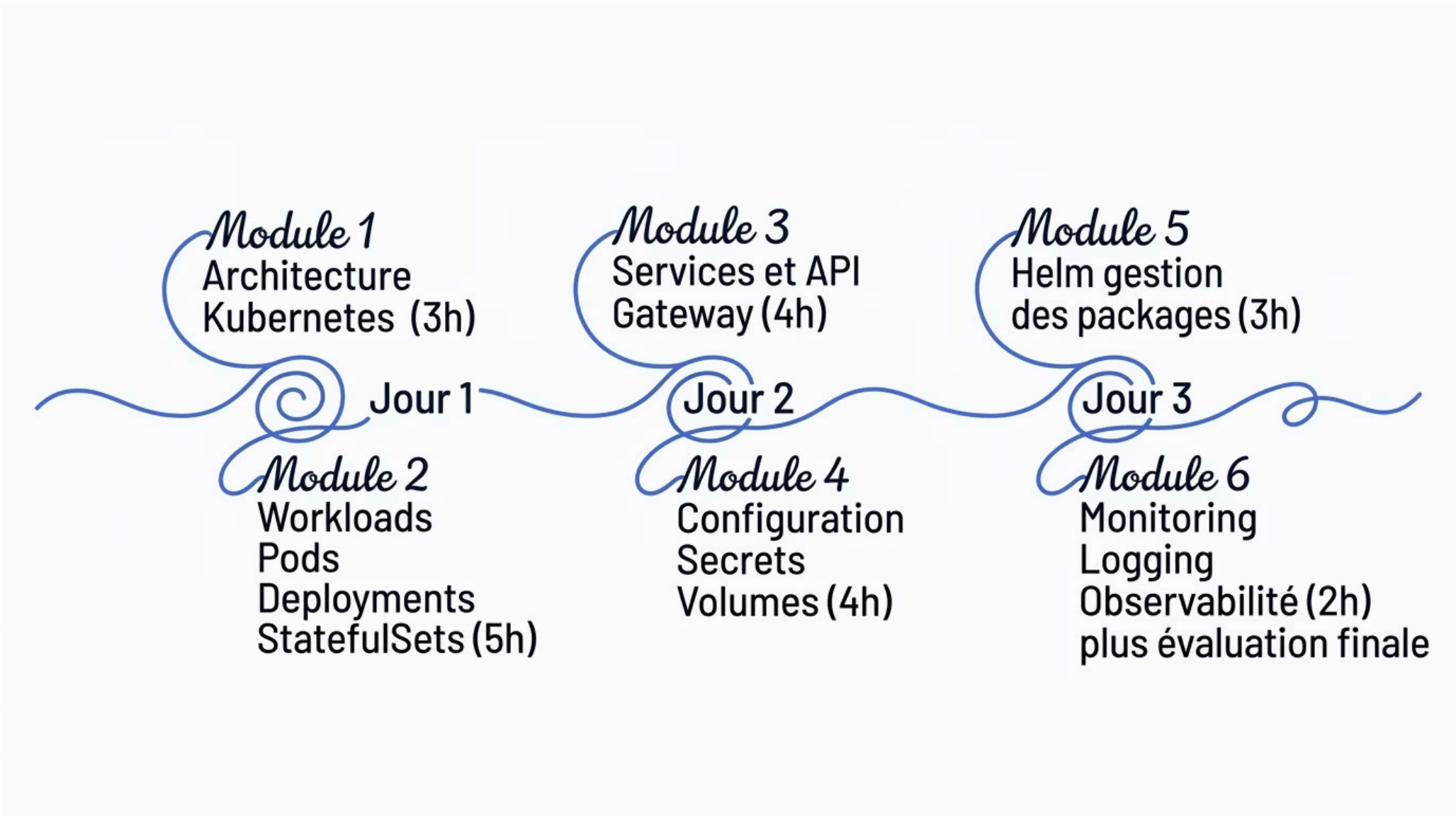
Centralisation des logs

Stack EFK (Elasticsearch, Fluentd, Kibana) ou Loki + Grafana pour agréger et interroger les logs de tous les pods.

Debugging en production

`kubectl top` pour les ressources, `kubectl events` pour les incidents, analyse des CrashLoopBackOff et OOMKill.

Les 3 jours en un coup d'œil



21h

De formation

Réparties sur 3 jours intensifs

6

Modules

Du fondamental à l'avancé

100%

Pratique

Ateliers kubectl, YAML et Helm

MÉTHODES & ÉVALUATION

Pédagogie et modalités d'évaluation

Méthodes pédagogiques

Cluster local sur poste

Chaque participant dispose d'un cluster Kubernetes local via **kind** sur sa propre machine.

Ateliers pratiques continus

Exercices avec `kubectl`, manifestes YAML et Helm tout au long des 3 jours.

Déploiement en mode projet

Application réelle issue du contexte des participants, déployée de A à Z.

Environnement post-formation

L'environnement reste disponible après la formation pour continuer à pratiquer.

Modalités d'évaluation

01

Exercices par module

Validation des acquis à chaque étape de la formation.

02

Déploiement complet

Déploiement d'une application complète avec rédaction d'un manifeste Helm.

03

Debugging simulé

Résolution de situations d'urgence : `CrashLoopBackOff`, `OOMKill`, quota atteint.

INFORMATIONS PRATIQUES

Financement, accessibilité et contact

Financement — Plan de Développement des Compétences

Cette formation est éligible au **Plan de Développement des Compétences (PDC)**. Plusieurs OPCOs peuvent prendre en charge tout ou partie du coût.

OPCO 2i	**Atlas**
Constructys	**Afdas**
AKTO	

Pour toute situation de **handicap**, contacter Telemach Learning avant l'inscription afin d'étudier les adaptations possibles.

Telemach Learning

24 Rue Chanzy
92600 Asnières-sur-Seine

SIRET	522 540 061 00010
NDA	11 92 27843 92
Email	contact@telemach-learning.fr
Web	www.telemach-learning.fr
Formateur	Alexandre BOIGUES

> Prêt à maîtriser Kubernetes ? Contactez-nous pour planifier votre session intra-entreprise.