

# Kubernetes dans un Contexte DevOps

## Objectif général :

Ce TP vise à familiariser les ingénieurs DevOps avec Kubernetes en leur permettant de mettre en pratique divers concepts et fonctionnalités de Kubernetes, allant du niveau débutant à avancé. Les participants seront amenés à déployer, gérer et mettre à l'échelle des applications conteneurisées en utilisant les outils et les commandes Kubernetes.

## Prérequis :

- Connaissance de base des conteneurs et de Docker
- Compréhension des concepts de base de Kubernetes

## Matériel requis :

- Accès à un cluster Kubernetes local ou en ligne
- Outils de ligne de commande Kubernetes (kubectl)

## Scénario :

Vous travaillez pour une entreprise de développement logiciel qui souhaite migrer ses applications vers une infrastructure basée sur des conteneurs. Votre équipe a été chargée de gérer cette migration en utilisant Kubernetes. Au cours de ce TP, vous allez travailler sur le déploiement d'une application Web (NodeJS fournie) dans un cluster Kubernetes. Créer au préalable une image de l'application fournie et envoyer cette image vers le DockerHub.

## Étapes du TP :

### 1. Installation et configuration de Kubernetes :

- Installez et configurez un cluster Kubernetes local ou en ligne.
- Vérifiez que le cluster est opérationnel en utilisant la commande kubectl.

### 2. Déploiement de l'application :

- Créez un déploiement Kubernetes pour l'application Web à l'aide d'un fichier de configuration YAML.
- Vérifiez que le déploiement est effectué correctement et que les pods sont en cours d'exécution.

### 3. Gestion des répliques :

- Mettez à l'échelle l'application en modifiant le nombre de répliques du déploiement.
- Vérifiez que les nouvelles répliques sont créées et déployées correctement.

### 4. Définition des services :

- Créez un service Kubernetes pour exposer l'application en interne.
- Vérifiez que le service est accessible à partir du cluster Kubernetes.

### 5. Exposition externe de l'application :

- Définissez un service de type LoadBalancer pour exposer l'application de manière externe.

- Vérifiez que l'application peut être accessible depuis l'extérieur du cluster en utilisant l'adresse IP attribuée.

#### 6. Gestion des mises à jour :

- Mettez à jour l'image de conteneur utilisée par l'application.
- Vérifiez que la nouvelle version de l'application est correctement déployée sans interruption de service.

#### 7. Surveillance et journalisation :

- Configurez la surveillance des ressources Kubernetes à l'aide de l'outil de votre choix (par exemple, Prometheus).
- Analysez et interprétez les journaux des pods de l'application à l'aide de l'outil de journalisation de votre choix (par exemple, Fluentd).

#### 8. Pipeline pour Déploiement sur un Cluster Kubernetes avec Jenkins

Proposer et Réaliser un pipeline CI/CD intégrant au moins Jenkins, Docker, Git, DockerHub, Kubernetes, Ansible et Nagios.