

Étude de Cas : Implémentation d'un pipeline CI/CD pour une application web déployée sur Kubernetes ou Docker

CFA INSTA
PARIS

1. Contexte

Vous êtes recrutés comme ingénieurs DevOps pour une entreprise de e-commerce. Votre mission est de mettre en place un pipeline CI/CD automatisé pour déployer une application web constituée de trois composants principaux :

1. **Frontend** : Une application React.js servie par Nginx.
2. **Backend** : Une API REST développée en Node.js.
3. **Base de données** : MongoDB, avec stockage persistant.

L'objectif est d'assurer des déploiements fiables, rapides, et sans interruption, tout en automatisant les étapes de test, construction, et déploiement.

2. Objectifs de l'Étude de Cas

Comprendre et appliquer les concepts de CI/CD.

Construire un pipeline complet pour tester, construire et déployer une application multi-conteneurs.

Déployer automatiquement les composants de l'application sur un cluster Kubernetes.

Mettre en œuvre des bonnes pratiques de configuration et d'automatisation.

3. Consigne

Technologies à utiliser :

1. Docker : Pour créer et gérer les conteneurs.
2. Kubernetes : Pour orchestrer et déployer les services.
3. GitLab CI/CD : Pour automatiser le pipeline de construction, test et déploiement.

Composants obligatoires :

1. Frontend : Déployé avec un service NodePort pour être accessible publiquement.
2. Backend : Déployé dans Kubernetes avec une communication interne avec MongoDB via un service ClusterIP.
3. Base de données MongoDB : Stockage persistant requis (PersistentVolume et PersistentVolumeClaim).

Pipeline CI/CD :

1. Une étape de Build pour construire les images Docker.
2. Une étape de Test pour vérifier le backend.
3. Une étape de Push pour envoyer les images vers Docker Hub ou un registre privé.

4. Une étape de Deploy pour appliquer les fichiers Kubernetes et mettre à jour les services.

4. Livrables

Fichiers de configuration :

1. Dockerfile pour chaque composant.
2. Fichiers YAML Kubernetes (Deployment, Service, PersistentVolume, etc.).
3. Fichier .gitlab-ci.yml pour configurer le pipeline CI/CD.

Preuve de fonctionnement :

1. Demonstration de fonctionnement

Documentation :

1. Description de l'architecture déployée.
2. Explication des étapes du pipeline CI/CD.

5. Étapes à suivre

1. Créer les Dockerfiles pour chaque composant :

Frontend (React.js) : Construire une image qui sert les fichiers avec Nginx.

Backend (Node.js) : Une API REST fonctionnelle.

MongoDB : Utiliser l'image officielle avec stockage persistant.

2. Créer les fichiers Kubernetes :

Déployer chaque composant dans un pod.

Configurer les services pour exposer le frontend (NodePort) et relier le backend à MongoDB (ClusterIP).

Ajouter un PersistentVolume et un PersistentVolumeClaim pour MongoDB.

3. Configurer GitLab CI/CD :

Ajouter un fichier .gitlab-ci.yml pour automatiser les étapes du pipeline.

4. Tester l'application :

Modifier le code du frontend ou du backend, et observer les changements déployés

automatiquement via le pipeline.