

# EC06 – CI/CD et versionning

## Contexte général

Dans le cadre du projet **RebootCamp**, cette épreuve consiste à mettre en œuvre une **chaîne d'intégration et de déploiement continu**, avec un système de versionning structuré.

L'objectif est de garantir la qualité, la traçabilité et la reproductibilité des livrables logiciels grâce à des outils DevOps. L'apprenant doit configurer un dépôt Git structuré, créer un pipeline CI/CD, conteneuriser l'application, et gérer le déploiement sécurisé sur un environnement cible, idéalement hébergé sur une infrastructure **éco-responsable**.

## Livrables attendus

Livrable	Détail
Dépôt Git	Organisation des branches ( main , dev , feature/... )
Pipeline CI/CD	Fichier de configuration (GitLab CI, GitHub Actions, Jenkins, etc.)
Dockerfile	Image conteneurisée de l'application
Script de déploiement	Automatisation du déploiement avec gestion des variables d'environnement
README	Documentation du processus complet : CI, Docker, déploiement

## Modalités d'évaluation

- **Type d'épreuve** : Mise en situation reconstituée sur ordinateur
- **Durée** : 4h
- **Nature** : Épreuve individuelle, sans oral
- **Critères évalués** :
  - C7.3 : Structuration du dépôt Git
  - C9.4 : Mise en place d'un pipeline CI/CD

- C9.5 : Déploiement sécurisé via conteneurisation

## Recommandations

- Utiliser une **stratégie Git claire** (Git Flow ou Trunk-Based) adaptée au travail collaboratif
- Versionner l'ensemble : code, scripts, configurations
- Construire un **Dockerfile fonctionnel** et optimisé (multi-stage, image légère en prod)
- Ajouter des étapes de **lint, tests, build** dans le pipeline
- Sécuriser les déploiements avec des **variables d'environnement sensibles** (tokens API, secrets DB)
- Prévoir un **fichier** `.env.dist` avec instructions pour l'exécution locale
- Documenter tout le processus en mettant en avant les **bonnes pratiques d'éco-conception DevOps** (images légères, pipelines sobres, réduction des builds inutiles)

## Exemples d'éléments attendus

- Un pipeline CI avec étapes : install → lint → test → build → deploy
- Un Dockerfile multistage produisant une image optimisée pour la production
- Un fichier `.env` , `.env.dist` et gestion sécurisée des secrets via le CI/CD
- Des captures ou logs de builds réussis sur GitHub Actions ou GitLab CI
- Un script `deploy.sh` automatisant le déploiement sur un serveur ou container cloud
- Mise en avant de solutions **éco-conçues** : limitation des jobs redondants, monitoring de l'impact énergétique

## Nom de dossier attendu

- une archive ZIP nommée `EC06_NomPrenom.zip` contenant :
  - Le dépôt complet (avec `.git` si demandé)
  - Le fichier `.gitlab-ci.yml` , `.github/workflows/ci.yml` ou équivalent
  - Le `Dockerfile` et les scripts associés ( `deploy.sh` , `run.sh` , etc.)
  - Un `README.md` expliquant le fonctionnement de la CI/CD

## Rappel pédagogique

Cette épreuve vise à valider la **maîtrise des pratiques DevOps essentielles** :

- Structuration et suivi des modifications via Git
- Automatisation des tâches de vérification, test et déploiement
- Mise en œuvre de **conteneurs Docker** pour homogénéiser les environnements
- Sécurisation et professionnalisation du processus de livraison
- Sensibilisation à l'**impact environnemental des pratiques DevOps**

Elle constitue une étape essentielle vers des pratiques d'**industrialisation logicielle modernes et durables**.