

# TP : Configuration d'un Pipeline CI/CD pour une Application Web PHP

Auteur : Christian Delettre

Date : 12/12/2023

## **Objectifs :**

- **Compréhension et mise en œuvre d'un pipeline CI/CD** : comprendre le projet PHP et être capables de configurer chaque composant du pipeline CI/CD.
- **Extension et personnalisation du pipeline** : ajouter de nouveaux jobs au pipeline CI/CD pour répondre à des besoins spécifiques.

## **Evaluation :**

Pour ce TP, vous formerez des groupes de 4 personnes. Ce TP se déroulera sur 2 séances de 3h.

A la fin de la deuxième séance, ou avant la date du 2<sup>ème</sup> TP + 6 jours (heure limite 23:59:59 UTC+2), vous m'enverrez par email un **fichier ZIP** contenant les livrables à l'adresse :

[teachings@deltekzen.com](mailto:teachings@deltekzen.com)

Un seul livrable par groupe est nécessaire.

## **Pour rappel :**

Tout élève qui ne m'envoie pas son livrable avant la date limite recevra une pénalité de -5 points sur la note finale. Aucune excuse ne sera acceptée.

La ligne d'objet de l'e-mail DOIT suivre ce modèle :

*TP01 SI5 devsecops – full\_name \_1 / full\_name \_2 / full\_name \_3 / full\_name \_4*

Une pénalité de -2 points sur votre note sera appliquée si vous ne suivez pas ce modèle.

**Pensez à mettre vos noms sur le rapport.**

## Objectifs

Le TP consiste en une application PHP utilisant Composer. Vous devrez mettre en place et étendre un pipeline CI/CD en utilisant GitHub, CircleCi, GHCR, Docker, Infisical, et AWS EC2.

## Parties

### Partie 1 : Compréhension et configuration de base

1. **Analyse du projet PHP** : Examiner le code source pour comprendre la structure et les fonctionnalités de l'application.
2. **Configuration de l'environnement Docker** : Comprendre et configurer Docker pour exécuter l'application localement.

### Partie 2 : Mise en place du pipeline CI/CD

1. **Configuration GitHub et CircleCi** : Créer le projet sur GitHub et configurer CircleCi.
2. **Création du pipeline CI/CD** : Comprendre et configurer les jobs CircleCi existants pour l'intégration continue, y compris la construction d'images Docker et le stockage sur GHCR. Configurer les variables nécessaires dans CircleCi.
3. **Gestion des secrets avec Infisical** : Configurer Infisical pour gérer les secrets de l'application de manière sécurisée.

### Partie 3 : Extension du Pipeline

1. **Ajout de jobs dévaluation de code** : Ajouter deux jobs d'évaluation du code automatique dans le pipeline : phpmetrics/phpmetrics et phploc par exemple.
2. **Intégration de la qualité du code** : Intégrer des outils de vérification de la qualité de code (par exemple, phpmd et niels-de-blaauw/php-doc-check) dans le pipeline.
3. **Déploiement automatisé sur AWS EC2** : Configurer le déploiement automatique sur AWS EC2, en tenant compte des meilleures pratiques de sécurité et de performance. Attention, le job de déploiement en production est manquant, à vous de jouer.

### Partie 4 : Documentation et Rapport

1. **Documentation du processus** : Rédiger une documentation expliquant chaque étape de la configuration et de l'extension du pipeline (Jobs, workflow(s)).

2. **Rapport final** : Présenter un rapport détaillé incluant les défis rencontrés, les solutions apportées, et les résultats obtenus. Un beau schéma récapitulatif du fonctionnement du pipeline CI/CD est plus que souhaitable.

### **Livrables :**

- Documentation complète du pipeline CI/CD.
- Code source du projet avec les fichiers de configuration mis à jour.
- Rapport détaillant l'ensemble du processus et les extensions apportées au pipeline.

### **Evaluation :**

- Compréhension et mise en œuvre correcte du pipeline CI/CD.
- Qualité et pertinence des extensions apportées au pipeline.
- Capacité à documenter et à expliquer le processus suivi et les choix effectués.