

Dans quelle mesure la  
modernisation technologique de la  
chaîne CI/CD, couplée à une  
stratégie de Knowledge  
Management, permet-elle de  
transformer l'obsolescence  
technique en levier stratégique ?

**Victorien GBAGUIDI**

Organisme d'accueil : **Société Générale CIB**  
Tuteur Pédagogique : **Khalid BELHAJJAME**  
Maître d'apprentissage : **Olivier CHANTEFOIN**

*Master 2 MIAGE SITN*  
Année Universitaire 2025–2026

# Introduction

Au sein de l'industrie informatique moderne, les chaînes d'intégration et de déploiement continues (CI/CD) jouent un rôle crucial dans la livraison rapide et fiable des logiciels. Cependant, avec l'évolution rapide des technologies et des pratiques de développement, ces chaînes CI/CD peuvent rapidement devenir obsolètes, posant des défis significatifs pour les entreprises d'un point de vue de l'efficacité opérationnelle et de la sécurité des systèmes d'information. Ce mémoire traitera la problématique suivante : **Dans quelle mesure la modernisation technologique de la chaîne CI/CD, couplée à une stratégie de *knowledge management* permet de transformer l'obsolescence technique en levier stratégique ?**

Ce mémoire propose une stratégie de migration de la chaîne CI/CD vers des solutions plus modernes en passant de Jenkins vers GitHub Actions et en faisant une transition de serveurs on-premises vers un cluster de serveurs Kubernetes. La migration technologique d'un microservice sera présentée dans cette étude. La solution comportera également une stratégie d'accompagnement au changement et de *knowledge management* pour pérenniser la connaissance au sein de l'entreprise et favoriser son adaptabilité.

## Contexte

Le service Alltra est en charge d'une application éponyme de gestion documentaire de transactions commerciales de matières premières. L'application est composée d'une quarantaine de microservices mais repose sur une infrastructure CI/CD vieillissante (Jenkins/XLDeploy). Cela pose des problèmes d'automatisation car, aujourd'hui, les microservices sont en partie déployés manuellement allongeant considérablement la mise en production. De plus, du fait d'une configuration vieillissante (Jenkinsfile / Groovy) la correction des erreurs de build est plus compliquée. Certains plugins Jenkins présentent en outre des failles de sécurité (Common Vulnerabilities and Exposures ou CVE) et ne sont plus maintenus par la communauté.

L'usage de GitHub Actions permettra de rationaliser la chaîne CI/CD. Aujourd'hui la redondance Jenkins/GitHub Actions crée un surcoût injustifié quant aux coûts des licences GitHub et au maintien des serveurs Jenkins tandis que GitHub et GitHub Actions forment une solution intégrée. De plus, d'un point de vue technique les compétences de l'équipe Alltra sont plus alignées avec l'usage de GitHub Actions (configuration YAML) qu'avec celui de Jenkins (configuration Groovy). La configuration YAML, déclarative, permet de faire abstraction de la logique sous-jacente de la configuration des pipelines CI/CD tandis que la configuration Groovy, impérative, force la programmation de la logique sous-jacente et augmente ainsi le risque d'erreurs et la complexité de la configuration.

Enfin, la connexion au serveur Jenkins depuis le serveur GitHub crée une complexité injustifiée dans la chaîne CI/CD. L'adoption de GitHub Actions supprime ce besoin de connexion et regroupe l'hébergement du code et son dé-

ploiement. Les plugins sur GitHub Actions sont appelés des actions. La Société Générale dispose d’actions internes sur GitHub Actions qui offrent des fonctionnalités d’automatisation CI/CD. Une gouvernance quant à la gestion du cycle de vie de ces actions sera nécessaire et discutée au cours de ce mémoire. La Société Générale peut également bénéficier d’un écosystème d’actions officielles venant de comptes certifiés (Verified Creators) contrairement aux plugins communautaires de Jenkins qui ne sont pas certifiés. Ma stratégie de migration consiste à exploiter les actions certifiées de GitHub Actions pour automatiser la chaîne CI/CD et à utiliser les plugins internes de la Société Générale pour ses besoins spécifiques.

Par ailleurs, les serveurs on-premises actuellement supportés seront décommissionnés cette année et doivent donc impérativement être remplacés. Je propose de migrer vers un cluster Kubernetes. Ce choix stratégique répond à trois enjeux :

1. **L’Immutabilité** : Contrairement aux serveurs on-premises historiques sujets à des conflits de configuration au fil des déploiements, l’usage de conteneurs éphémères assure une isolation de l’environnement d’exécution en isolant chaque microservice et limite ainsi les effets de bord. Les conteneurs permettent également de conserver la même configuration sans impacter l’infrastructure matérielle sous-jacente. Cela limite les besoins de maintenance au niveau de la configuration et simplifie la maintenance de l’infrastructure CI/CD. Les conteneurs garantissent également une configuration dite iso-prod assurant que les dépendances du code source en environnement de développement soient identiques à celles en environnement de production.
2. **L’Élasticité** : La capacité d’*autoscaling* permet d’adapter dynamiquement les ressources à la charge réelle, assurant la performance sans intervention humaine.
3. **L’Optimisation des ressources** : En lançant plusieurs microservices sur un même serveur à travers des conteneurs sur un cluster Kubernetes au lieu de les déployer isolément sur des serveurs distincts, on optimise l’usage des ressources et on réduit les coûts de maintenance des serveurs existants ainsi que la complexité de l’infrastructure informatique.

Enfin, nous démontrerons que la réponse à l’obsolescence n’est pas uniquement technologique, mais culturelle. La migration vers des standards modernes impose une **conduite du changement** rigoureuse pour accompagner la montée en compétence des équipes vers une culture *DevOps* autonome.

De plus, afin d’éviter que la nouvelle solution ne devienne la dette technique de demain, nous aborderons les stratégies de *knowledge management*. À travers l’adoption de pratiques *InnerSource* et la documentation *as-code*, l’objectif est de pérenniser le savoir-faire et de garantir la maintenabilité corrective et évolutive à long terme de la chaîne de livraison.