Chapitre 2 : Automatisation du déploiement de templates NiFi sur les instances sur des pods Kubernetes

Introduction :

Comme mentionné dans la problématique ci-dessus, le problème du déploiement manuel répétitif des templates NiFi prend beaucoup de temps et laisse une très grande marge pour l’erreur humaine. Ceci est très palpable quand on découvre l’importance des templates NiFi dans le travail journalier des Data Engineers. Ces derniers utilisent les templates NiFi afin de ne pas avoir à faire et refaire les mêmes DataFlows qui s’avèrent parfois très longs et qui prennent beaucoup de temps à composer et à paramétrer. Il y a plusieurs exemples de DataFlows qui se composent de plusieurs ProcessGroups et de plusieurs Processors. En total, ceci peut arriver à des centaines de composantes qui sont prédéfinis. Ceci facilite largement le travail des Data Engineers qui n’ont donc plus qu’à changer les paramètres de contexte et personnaliser le Data Flow si cela s’avère nécessaire.

On peut donc dire que les templates NiFi aident les Data Engineers à être plus productifs et à avoir plus de temps à consacrer pour des problèmes plus difficiles. En plus d’avoir des templates prêts à être utilisés qui sont partagés entre les Data Engineers de la société dans un Repository dédié aux templates. Ces templates (en format xml) sont alors déployés par les Data Engineers sur des instances de NiFi sur les NiFiCluster de GKE (Google Kubernetes Engine) qui sont mis en disposition des Data Engineers.

Pour faciliter encore plus la chose et accélérer le déploiement des templates, nous avons implémenté un pipeline afin d’automatiser le téléchargement et l’instanciation des templates NiFi sur les instances NiFi au sein de Minikube. Pour ce faire, nous avons utilisé une panoplie d’outils et de technologies proposés par notre encadrant. La première partie de ce chapitre va couvrir alors les technologies utilisées lors de notre projet et qui ont été d’une grande utilité

Afin de concevoir et d’implémenter notre pipeline, nous avons tout d’abord eu à simuler l’environnement GKE sur nos machines locales afin de tester et améliorer notre solution pour qu’elle réponde au mieux au besoin exprimé par les Data Engineers. Pour cette simulation de l’environnement de Production, nous avons utilisé Minikube et l’opérateur Nifikop pour le déploiement d’un NifiCluster qui nous a servi comme environnement de test. La deuxième partie de ce chapitre va couvrir comment on a pu simuler l’environnement de Production le plus proche à la réalité afin de pouvoir concevoir et implémenter la solution la plus réaliste possible à notre problématique.

La troisième partie de ce chapitre va couvrir l’architecture qu’on a choisie pour notre pipeline et qui comporte les outils et technologies mentionnés précédemment. Le diagramme d’architecture dans cette partie montre comment les outils mentionnés sont utilisés en synergie afin de garantir la bonne implémentation du pipeline. Nous avons également consacré une partie pour l’implémentation de notre solution qui montre des captures d’écran exhibant comment on a pu déployer automatiquement et juste par un le déclenchement de notre pipeline un ensemble de templates (en format xml) vers une instance NiFi sur notre cluster Minikube tout en gardant une traçabilité pour la date du déploiement et les templates déployés.

Conclusion :

Comme on a vu dans ce chapitre, nous avons examiné en détail les différentes facettes de notre approche visant à résoudre le problème du déploiement manuel et répétitif des templates NiFi. À travers une exploration approfondie des outils, des méthodes de simulation d'environnement de production et de l'architecture mise en place pour notre pipeline, nous avons éclairé les étapes clés de notre solution.

Nous avons identifié les contraintes inhérentes au déploiement manuel des templates NiFi, mettant en lumière la perte de temps et le risque d'erreur humaine qui en découlent. La valorisation des templates NiFi en tant qu'élément essentiel du flux de travail quotidien des Data Engineers a été soulignée, montrant comment ils permettent d'améliorer la productivité en automatisant les flux de données récurrents. De plus, en mettant en évidence le partage de templates dans un référentiel dédié, nous avons montré comment cette approche favorise la collaboration et la réutilisation au sein de l'équipe.

Nous avons ensuite abordé la simulation de l'environnement de production, une étape cruciale pour concevoir et tester notre solution. L'utilisation de Minikube et de l'opérateur Nifikop a permis de recréer fidèlement l'environnement GKE, offrant ainsi une plate-forme de développement réaliste pour itérer et perfectionner notre pipeline.

L'architecture détaillée de notre pipeline a été présentée, mettant en évidence la synergie entre les divers outils et technologies que nous avons utilisés. Le diagramme d'architecture a clairement illustré comment ces éléments s'associent pour garantir la mise en œuvre réussie du pipeline, automatisant le téléchargement et l'instanciation des templates NiFi sur les instances NiFi de notre cluster Minikube.

L'implémentation de notre solution a été abordée, avec des captures d'écran fournissant des preuves tangibles de la réalisation de notre objectif. Nous avons démontré comment notre pipeline permet non seulement le déploiement automatisé des templates NiFi, mais assure également la traçabilité des déploiements, offrant ainsi une visibilité essentielle pour le suivi et la gestion.

En somme, ce chapitre a mis en lumière notre démarche méthodique pour résoudre efficacement la problématique du déploiement des templates NiFi. Les outils sélectionnés, la simulation de l'environnement de production et l'architecture bien pensée de notre pipeline convergent pour offrir une solution pratique, efficiente et traçable, répondant aux besoins cruciaux des Data Engineers et contribuant ainsi à l'amélioration globale de la gestion des flux de données au sein de notre environnement opérationnel. Le prochain chapitre mettra en évidence les résultats obtenus grâce à cette solution novatrice et leur impact sur l'efficacité opérationnelle de notre équipe.