Conception et Architecture du pipeline :

Afin d’implémenter notre pipeline, nous avons utilisé plusieurs outils et services qui nous ont permis de garantir le bon déroulement des étapes et l’automatisation de tout le processus de déploiement. Les outils que nous avons utilisés sont : GitLab, Terraform, Docker, Minikube, Nifikop et Nifi.

GitLab

.gitlab-ci.yml

Machine Locale

Docker

Minikube

Nifikop

Zookeeper Cluster

Pod

Pod

Pod

NifiCluster

NifiPod

nifi\_templates\_deploy.py

Terraform

main.tf

Comme illustré en détail dans le schéma architectural présenté ci-dessus, notre pipeline trouve son point de départ dans l'environnement de GitLab. Ce mécanisme s'enclenche sur notre machine locale avec GitLab Runner, qui orchestre habilement les actions requises. La raison d’avoir utilisé GitLab Runner est pour avoir accès à notre cluster Minikube local afin de tester le pipepline avant de travailler sur GKE.

Poursuivant le cheminement du processus, nous rencontrons le script principal Terraform, baptisé "main.tf", jouant un rôle crucial. Ce script ne se contente pas uniquement de garantir le bon fonctionnement de Minikube, mais il agit également en tant que catalyseur pour l'exécution du script python "nifi\_templates\_deploy.py".

Le script python en question joue alors son rôle, interagissant avec l’API Rest offerte de NiFi afin d’uploader le template originalement déposé comme fichier *xml* et l’instancier sur le pod Nifi en question. Ce pipleine permet le déploiement des modèles (templates) spécifiés.

**Implémentation :**

Pour l’implémentation de notre architecture, nous avons suivi les étapes suivantes :

1. Préparation du script python *nifi\_templates\_deploy.py* qui est le script principal de notre pipeline et qui va interagir directement avec l’API Rest de Nifi. Ce script se compose de 3 parties :
2. Récupérer du Token d’accès
3. Uploader les fichier xml du templates spécifiés
4. Instancer des ProcessGroup pour les templates uploadés

Ces trois étapes étaient faites manuellement et pour chaque template ce qui posait un problème pour les data engineers puisque ça prend beaucoup de temps surtout pour les templates qu’on veut instancier plusieurs fois. Faire cette tâche manuellement peut donner une grande marge d’erreur humaine qu’on peut éviter. C’est donc une problématique qui diminue la productivité des collaborateurs et met en risque le bon travail des dataflows NiFi.

1. Préparation du script *main.tf*:

Pour ce script, il a pour rôle de s’assurer que le cluster Minikube est bien en marche et qu’on pourra procéder pour l’exécution du script *nifi\_templates\_deploy.py*. Il ajoute également une autre couche d’abstraction sur le script python.

1. Création du pipeline sur GitLab avec le fichier *.gitlab-ci.yml*:

Pour ce pipeline, nous avons listé les étapes à accomplir par GitLab Runner afin d’exécuter le script *main.tf* sur notre machine locale.