

Rapport d'Analyse de la Cloud Function pour l'Examen

Ayoub Soukrat

28 mars 2025

1 Introduction

Ce document présente l'analyse détaillée du projet de Cloud Function développé sur Google Cloud Platform (GCP) en Node.js/TypeScript. L'objectif est de traiter un fichier de coordonnées et un fichier GeoJSON afin d'identifier les points situés à l'intérieur des polygones et de générer un fichier CSV (`resultat.csv`) contenant, pour chaque point, des informations supplémentaires provenant du polygone correspondant.

2 Description des Fichiers Utilisés

- **coordonnees.txt** : Fichier texte contenant les coordonnées géographiques. Le fichier est organisé sous forme tabulaire avec un en-tête (par exemple, `UID`, `Latitude`, `Longitude`) et chaque ligne représente un point.
- **plan-culture.geojson** : Fichier GeoJSON contenant une collection de polygones. Chaque polygone possède plusieurs propriétés décrivant les informations agricoles, telles que `clecomposite`, `nochamp`, `culture`, `variete`, `nosemi`, `date_semi`, etc.

3 Pseudocode de la Fonction Cloud Function

```
1 Debut :
2   1. Lire le fichier "coordonnees.txt" depuis Cloud Storage.
3     - Ignorer la premiere ligne (en-tete).
4     - Pour chaque ligne, extraire UID, Latitude, et Longitude.
5     - Convertir les valeurs decimales (remplacer la virgule par un
      point).
6   2. Lire le fichier "plan-culture.geojson" depuis Cloud Storage et le
      parser en objet GeoJSON.
7   3. Pour chaque point extrait :
8     a. Creer un objet GeoJSON de type Point avec les coordonnees [
        Longitude, Latitude].
9     b. Pour chaque polygone dans le GeoJSON :
10      - Verifier si le point est contenu dans le polygone (
        booleanPointInPolygon).
11      - Si oui, extraire les proprietes du polygone (clecomposite,
        nochamp, culture, variete, nosemi, date\_semi).
```

```

12         - Enregistrer la correspondance (UID, Latitude, Longitude et
13           proprietes).
14         - Passer au point suivant.
14     4. Generer un fichier CSV "resultat.csv" avec une ligne d'en-tete et
       toutes les correspondances.
15     5. Sauvegarder le fichier CSV dans le bucket Cloud Storage.
16     6. Retourner une reponse HTTP indiquant le nombre de resultats trouves
17     .
17 Fin.

```

Listing 1 – Pseudocode de la fonction processCoordinates

4 Analyse des Données du Fichier plan-culture.geojson

Le fichier GeoJSON contient une collection de polygones associés à des informations agricoles. Voici la signification des colonnes utilisées (les colonnes `fill` et `fill-opacity` sont ignorées) :

- **fid** : Identifiant de la feature, attribué automatiquement.
- **clecomposite** : Code composite unique, combinant l'année, le numéro de champ et un code de semis (ex : 2024-A200-B207).
- **uuid** : Identifiant unique universel.
- **organisme** : Nom de l'organisme responsable (ex : IRDA).
- **annee** : Année d'exploitation ou de semis.
- **nochamp** : Numéro ou identifiant du champ.
- **culture** : Type de culture pratiquée (ex : Blé printemps).
- **variete** : Variété de la culture (peut être `None` si non renseignée).
- **noaleatoire** : Numéro aléatoire ou code interne.
- **nosemi** : Code ou numéro de semis.
- **date_semi** : Date du semis.
- **idUnique** : Identifiant unique additionnel.
- **Note** : Remarques ou informations complémentaires.
- **stroke** : Couleur de la bordure du polygone (ex : #D9D9D9).
- **stroke-width** : Épaisseur de la bordure.
- **stroke-opacity** : Opacité de la bordure.
- **idElement** : Identifiant de l'élément, utilisé pour catégoriser la zone.
- **geometry** : Contient la géométrie du polygone (type `Polygon` avec des coordonnées sous la forme [Longitude, Latitude]).

5 Résultats Obtenus

Après exécution de la Cloud Function, le fichier `resultat.csv` a été généré et contient 4 résultats. Ces résultats indiquent que 4 points ont été identifiés comme étant situés à l'intérieur des polygones. Les points se trouvent dans une région spécifique, par exemple autour de Montréal, avec des coordonnées approximatives de 45.55°N et -73.36°W.

Chaque ligne du fichier CSV inclut :

- **UID** : Identifiant du point.
- **Latitude** et **Longitude** : Coordonnées du point.

- **clecomposite**, **nochamp**, **culture**, **variete**, **nosemi**, **date_semi** : Informations extraites du polygone dans lequel le point se trouve.

6 Tests Unitaires et de Déploiement

Afin de valider le bon fonctionnement de la Cloud Function, plusieurs tests ont été réalisés :

6.1 Tests Unitaires

Des tests unitaires ont été effectués pour vérifier :

- Le parsing du fichier `coordonnees.txt` :
 - Exemple : Une ligne comme `"abcg4g 45,5534 -73,3600"` doit être convertie en l'objet `{uid: "abcg4g", lat: 45.5534, lon: -73.3600}`.
- La conversion correcte des décimales (remplacement de la virgule par un point).
- La logique d'intersection des points avec les polygones à l'aide de la fonction `booleanPointInPolygon`.

6.2 Tests de Déploiement

La fonction a été déployée sur GCP et testée via :

- L'appel HTTP de la Cloud Function via `curl` pour vérifier la réponse et la génération du fichier `resultat.csv`.
- La vérification manuelle dans le bucket Cloud Storage pour s'assurer que le fichier CSV contient les bonnes données.

7 Visualisation des Données

Une image a été générée pour illustrer les résultats. Dans cette visualisation :

- Les polygones sont affichés avec une coloration basée sur le numéro de champ (**nochamp**).
- Les points identifiés sont superposés aux polygones.
- La carte est affichée en mode satellite pour une meilleure contextualisation géographique.

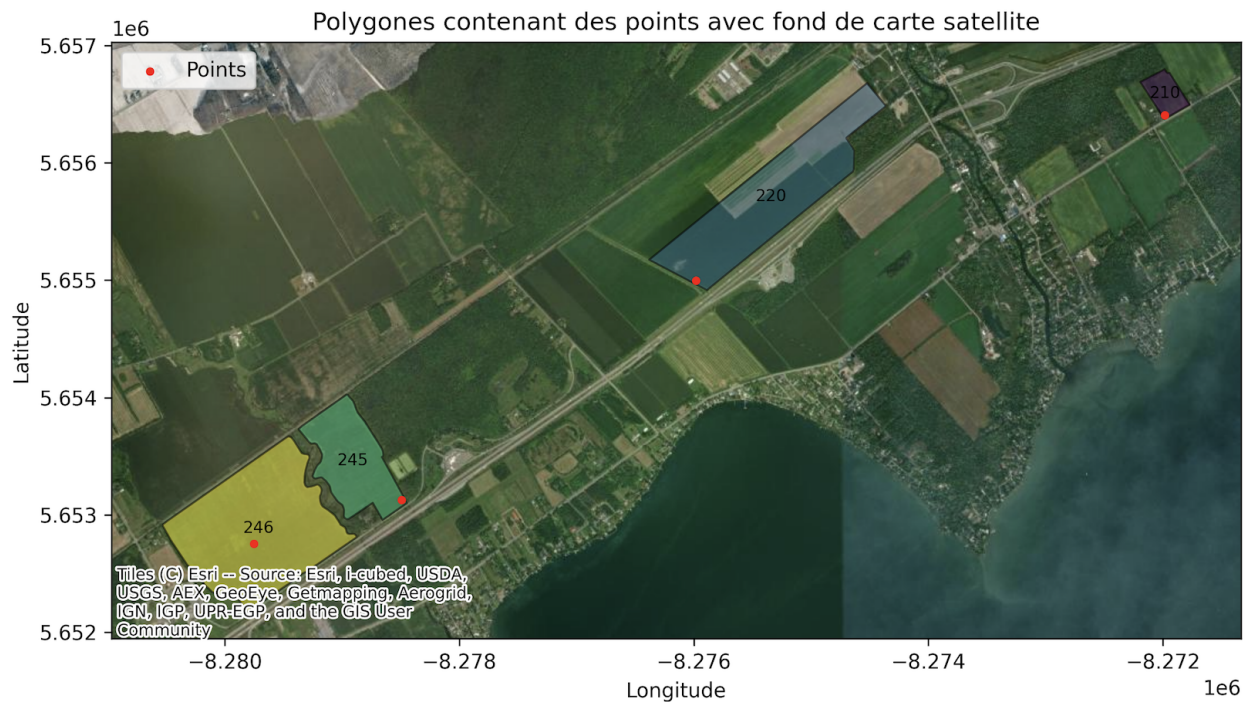


FIGURE 1 – Visualisation des points et des polygones sur une carte satellite, colorés selon le numéro de champ.

8 Conclusion

Ce projet a permis de développer et de déployer une Cloud Function efficace sur GCP pour analyser des données géospatiales. Le fichier `resultat.csv` généré fournit des informations détaillées sur chaque point se trouvant dans les zones définies par le fichier GeoJSON. Des tests unitaires et de déploiement ont validé la robustesse de la solution. La visualisation graphique offre une compréhension visuelle du résultat.

Des pistes d'amélioration futures incluent l'optimisation du traitement des données, l'ajout de tests automatisés supplémentaires et l'extension de la visualisation interactive.