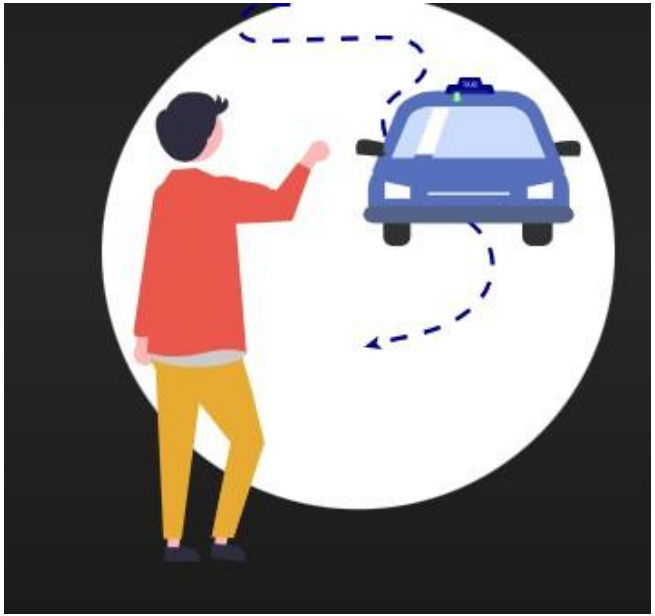
Projet Big Data Streaming

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE



Contexte du projet

Ce projet vise à développer le backend d'un outil permettant de calculer la distance entre un chauffeur de taxi et un client afin de déduire le prix du trajet selon le confort choisi.

Il permet également de mettre en place des tableaux de bord et un Datawarehouse pour faire du machine learning en temps réel

Architecture Backend

1. **Python** pour l'intégration des données dans Kafka
2. **Nifi** pour la transformation des données et l’ingestion dans Elasticsearch et GCS
3. **Elasticsearch** pour l'indexation des données
4. **Kibana** pour la visualisation des données
5. **Google Cloud Storage** pour le stockage externe des table Big Query
6. **Big Query** pour le Datawarehouse

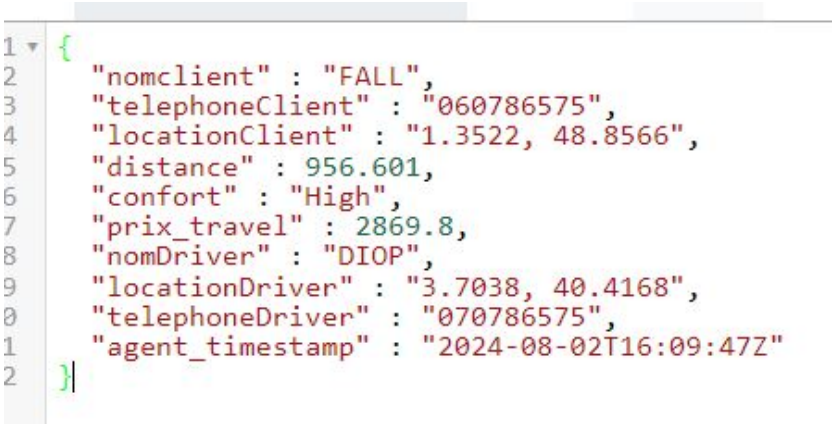
Modèle de données entrant dans Kafka


```
{
    "data": [
        {
            "confort": "standard",
            "prix_base_per_km": 2,
            "properties-client": {
                "logitude": 2.3522,
                "latitude": 48.8566,
                "nomclient": "FALL",
                "telephoneClient": "060786575"
            },
            "properties-driver": {
                "logitude": 3.7038,
                "latitude": 40.4168,
                "nomDriver": "DIOP",
                "telephoneDriver": "070786575"
            }
        }
    ]
}
```


https://github.com/idiattara/Spark_DIATTARA/blob/main/data_projet.json

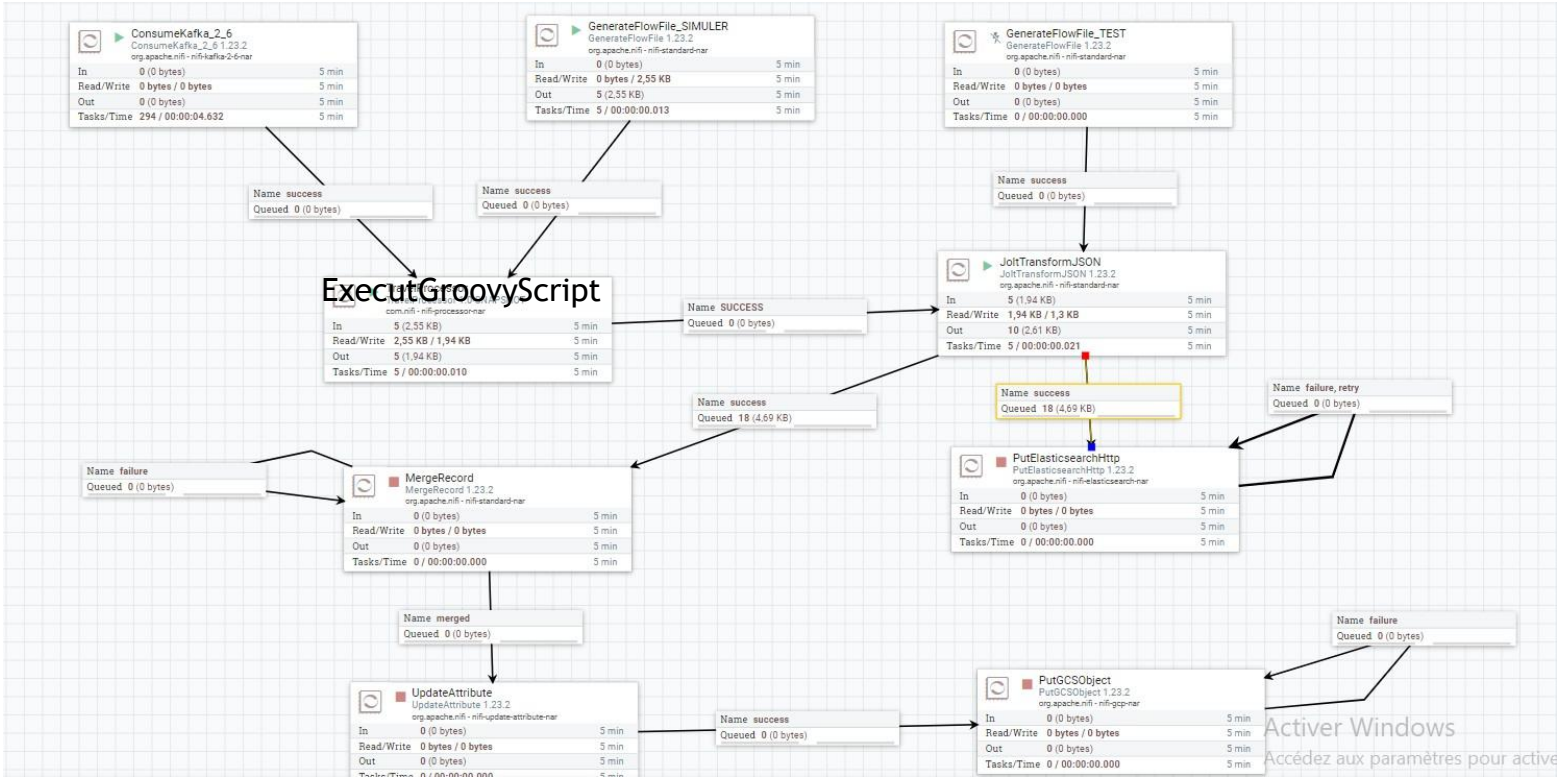
Exemple de sortie du processor ExecutGroovyscript

Exemple de sortie après le processor JoltransformJSON

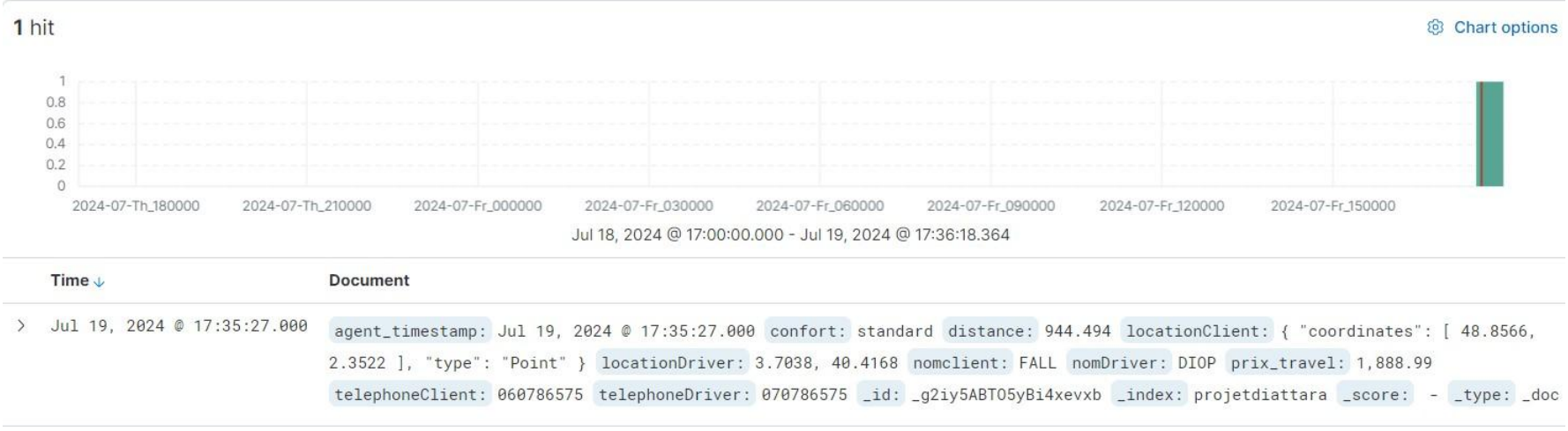


```
{
  "nomclient" : "FALL",
  "telephoneClient" : "060786575",
  "locationClient" : "1.3522, 48.8566",
  "distance" : 956.601,
  "confort" : "High",
  "prix_travel" : 2869.8,
  "nomDriver" : "DIOP",
  "locationDriver" : "3.7038, 40.4168",
  "telephoneDriver" : "070786575",
  "agent_timestamp" : "2024-08-02T16:09:47Z"
}
```

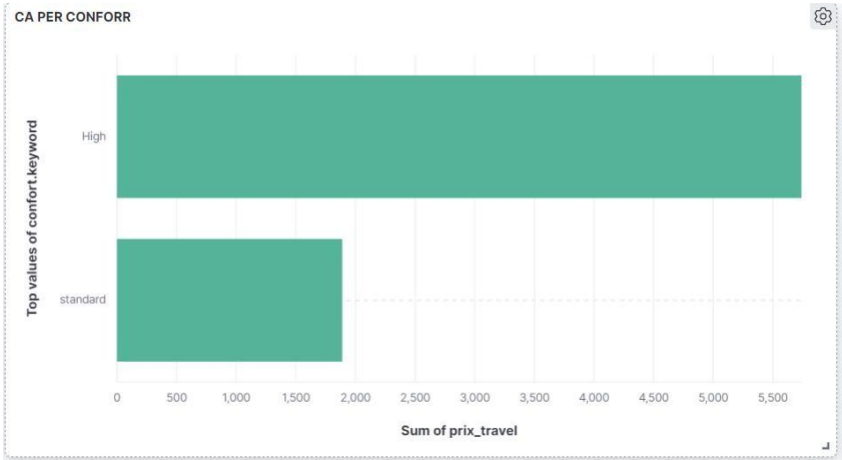
Urbanisation du workflow nifi

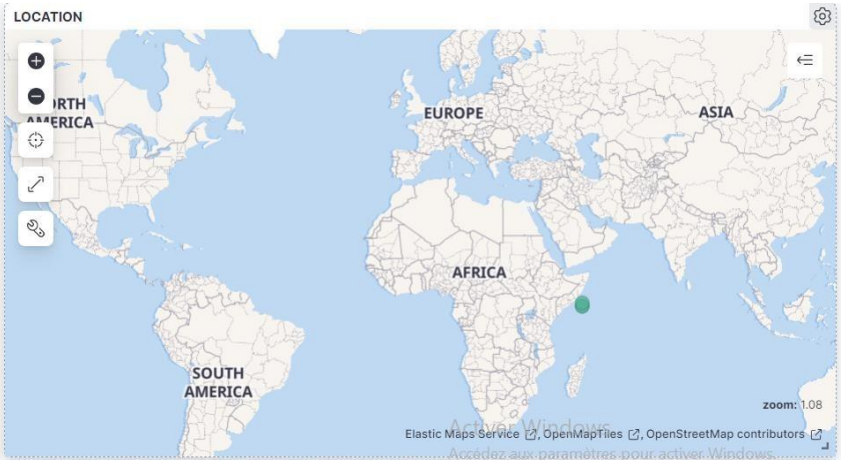


Exemple de Discover sous Kibina



Dashboard sous kibana





NIFI/Datawarehouse

Utiliser le Processor MergeRecord de Nifi pour créer des fichers avec 10000 reccord au format parquet mais pour tester vou
pouvez utilisez que 2 reccord

Dans votre GCS les fichier doivent avoir le format: ${now():format("yyyy-MM-dd'T'HH:mm:ss'Z'", "GMT")}.parquet

BigQuery ML

Créer une **Table External** avec **DataWarehouse (BigQuery)** qui pointe vers votre le répertoire de votre **bucket cloud storage**

Utiliser le **fichier CSV** fourni par le client afin de créer **8 cluster** avec les variables longitude et latitude en utilisant **K-Means** https://github.com/idiattara/Spark_DIATTARA/blob/main/uber-split2.csv

Calculer en temps réel le chiffre d'affaire de chaque cluster pour chaque type de confort(hight, Medium, low, ..) des data présentes dans votre **Dalake-Lak**(cloud storage)

Architecture & Budget

Proposer une **architecture** de votre plateforme (Disk, RAM, CPU, partition , …) => Kafka , Elastic

Estimer le coût du projet pour la patie Google cloud

prix
-