TP 5 : Intégration de Données d'une API vers Elasticsearch

Cours EFREI 2024-2025

Yvann VINCENT
Machine Learning Engineer

Introduction

Dans ce TP, vous allez créer un pipeline de données qui extrait des données brutes depuis l'API **HackerNews** et les insère dans un cluster **Elasticsearch**. Ce TP introduit les concepts d'intégration d'API et de stockage dans un moteur de recherche distribué.

Le workflow général sera structuré comme suit :

- Extraction des données brutes via l'API HackerNews.
- Transformation minimale des données pour structurer les entrées.
- Insertion des données dans Elasticsearch.

Un `Dockerfile` et un `docker-compose.yml` ont déjà été fournis pour configurer votre environnement.

—

1 Exercice 1 : Configuration de l'Environnement

1.1 Objectif

Mettre en place l'environnement nécessaire pour intégrer les données d'une API dans Elasticsearch.

1.2 Étapes

1. Vérifiez que Docker et Docker Compose sont installés sur votre machine.
2. Lancez les conteneurs avec :

   ```
   docker-compose build
   docker-compose up -d
   ```

3. Testez l'accès à Elasticsearch en ouvrant `http://localhost:9200` dans votre navigateur ou avec la commande suivante :

   ```
   curl -X GET http://localhost:9200
   ```

   Si tout fonctionne, vous devriez voir une réponse JSON indiquant que le cluster est en ligne.

4. Installez les dépendances Python nécessaires :

   ```
   pip install requests elasticsearch
   ```

—

2 Exercice 2 : Extraction des Données Brutes

2.1 Objectif

Utiliser l'API HackerNews pour récupérer les dernières actualités au format JSON.

2.2 Étapes

1. Consultez la documentation de l'API HackerNews (`https://github.com/HackerNews/API`) pour comprendre les endpoints disponibles.
2. Récupérez les 50 dernières actualités en utilisant l'endpoint `/v0/topstories.json`. Ce dernier retourne une liste d'identifiants.
3. Pour chaque identifiant, utilisez l'endpoint `/v0/item/<ID>.json` pour récupérer les détails de l'article.
4. Stockez les réponses JSON dans le bucket raw
5. Attention, si vous avez nettoyé votre environnement virtuel avec un docker system prune ou similaire, vous allez devoir re-créer les buckets.

Note : Créez un script Python avec un `argparse` pour spécifier le nombre de nouvelles à extraire.

—

3 Exercice 3 : Transformation et Indexation dans Elasticsearch

3.1 Objectif

Transformer les données JSON brutes et les insérer dans un index Elasticsearch.

3.2 Étapes

1. Créez un index nommé `hackernews` dans Elasticsearch. Utilisez les commandes suivantes pour définir un mapping :

```
curl -X PUT "http://localhost:9200/hackernews" -H 'Content-Type: application/json' -d'
{
  "mappings": {
    "properties": {
      "id": { "type": "integer" },
      "title": { "type": "text" },
      "content": { "type": "text},
      "url": { "type": "keyword" },
      "score": { "type": "integer" },
      "timestamp": { "type": "date" }
    }
  }
}
'
```

2. Écrivez un script Python pour transformer les données JSON. Assurez-vous que chaque document à insérer contient les champs suivants :
   - **id** : L'identifiant de l'article.
   - **title** : Le titre de l'article.
   - **url** : L'URL de l'article.
   - **score** : Le score de l'article.
   - **timestamp** : La date de création (convertie en format ISO 8601).
3. Insérez les documents transformés dans l'index `hackernews` à l'aide du client Elasticsearch de Python.
4. Vérifiez que les données ont bien été insérées en utilisant la commande :

```
curl -X GET "http://localhost:9200/hackernews/_search?q=*&pretty"
```

—

4 Exercice 4 : Extension et Optimisation (Facultatif)

4.1 Objectif

Étendre et optimiser le pipeline en :

- Créez un DAG dans le fichier dags/hackernews.py afin d'associer la récupération de données depuis l'API et l'insertion dans le base Elasticsearch
- Paramétrez le DAG de façon à se ce qu'il se trigger toutes les 5 minutes
- Lancez le depuis l'interface de Airflow

—

Conclusion

À la fin de ce TP, vous aurez appris à :

- Extraire des données depuis une API publique.
- Configurer et interagir avec Elasticsearch pour indexer et rechercher des données.
- Créer un pipeline de données complet avec extraction, transformation et chargement.