T-DAT-901 - Crypto Viz

WikiMedia
Report

Hugo Vincent
Cao-Thanh-Uy Nguyen
Arnaud Troise
Arnaud Gadbin

Github repository

1. Introduction

1.1 - Aperçu

WikiMedia est une solution technique conçue pour aider les utilisateurs à collecter, traiter et visualiser efficacement les données et les tendances de Wikipédia en temps réel.

1.2 - Problématique

Wikipédia est une plateforme ouverte, et en constante évolution, où des milliers de modifications sont effectuées chaque jour, sur de nombreux articles. Le traitement et la visualisation des données en temps réel sont essentiels pour analyser l'évolution de ces articles, détecter les tendances et repérer d'éventuelles activités suspectes.

D'après ce blog poste de stanford, wikipedia a développé des points à surveiller pour lutter contre le vandalisme et les comportements suspects. Notre but sera d'en implémenter plusieurs, notemment:

- **Niveaux élevés d'activité d'édition par rapport à la notoriété** . Il y a 1,9 édition par seconde sur Wikipédia ; cela équivaut à un peu moins d'une édition par page et par mois. Si une page a un niveau d'activité nettement plus élevé, il vaut la peine d'en chercher la raison. Il est possible que vous ayez trouvé une guerre d'édition (mais elle pourrait être boiteuse).

- **Pages avec de nombreuses modifications de la part d'un ou de quelques utilisateurs** . Si la grande majorité des modifications d'une page proviennent d'un petit nombre d'utilisateurs sur une période prolongée, il se peut qu'il y ait un problème. Bien qu'il soit courant que les éditeurs aient une expertise en la matière (et soient donc très actifs dans certains réseaux de pages) et/ou assument la « responsabilité » des pages qu'ils ont créées, les éditeurs trop actifs méritent une enquête plus approfondie.

Nos objectifs sont donc généralement de tracker les niveaux d'éditions d'articles; pour voir les articles à surveiller, le nombre de contributeurs; pour voir ceux à surveiller, et si possible lier ces informations par rapport au trafic sur les articles.

1.3 - Objectifs

L'objectif principal de ce projet est de fournir une solution complète pour l'analyse des données de Wikipédia, avec l’accent sur le processus et la démarche Data Analyste . Les objectifs sont les suivants :

- Collecte continue et en stream des données des pages Wikipédia et des modifications récentes.
- Traitement en temps réel des données.
- Visualisation dynamique des tendances et chiffres à l’aide d’une interface graphique définie avec des maquettes.

1.4 - Portée du projet

Ce projet couvre la conception et le déploiement d’une architecture Kafka, avec une récupération de données en stream, un traitement de données adapté à notre problématique et une visualisation finale des données par une interface graphique.

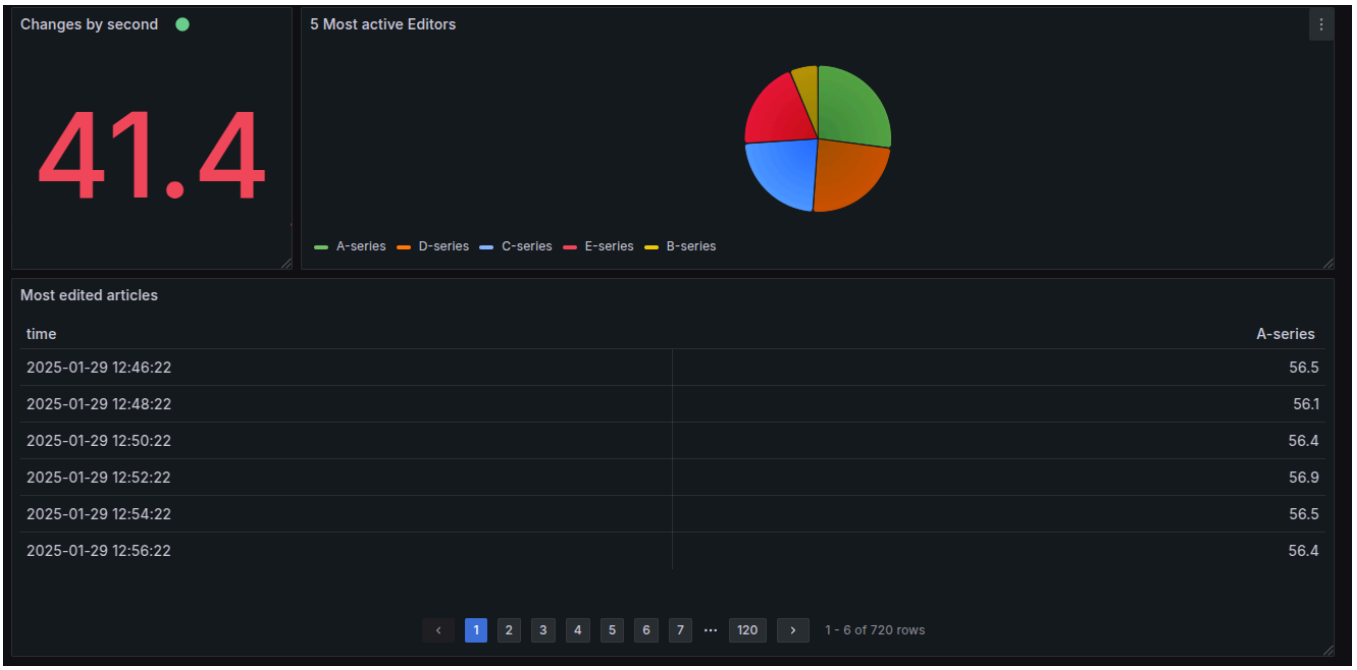
2. Maquettage

En commençant par le maquettage, on s'assure d'avoir des objectifs clairs et un but à accomplir.

Pour ce faire, on à développé une maquette, en sachant que l'on va utiliser Grafana pour l'affichage, nous permettant d'avoir une idée générale de départ sur le design.

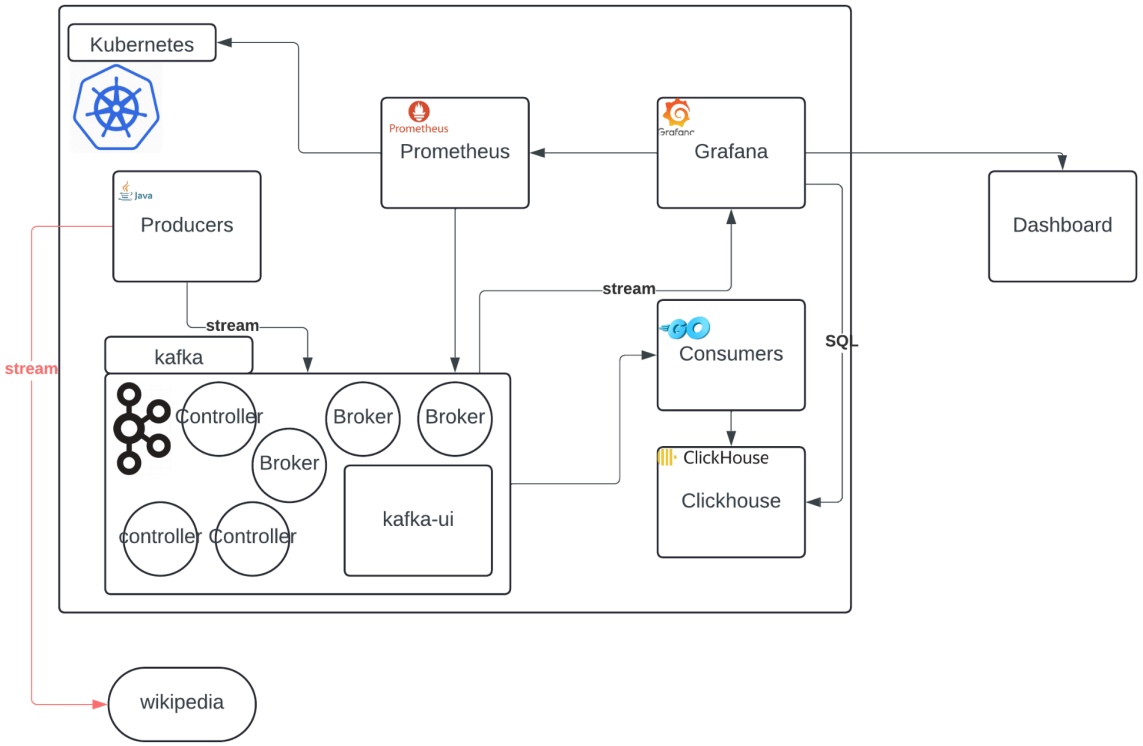
L'outil étant simple et modulable, la maquette était vouée à changer au moins un peu, après l'implémentation technique, et les retours des premières utilisations.

Nous avons élaboré cette maquette directement sur l'outil.



3. Architecture du Système et Choix Techniques

- **Langages de Programmation** : Java (producers, consummers).
- **Messagerie et Streaming** : Apache Kafka.
- **Stream** : KafkaStreams.
- **Base de Données** : ClickHouse..
- **Déploiement** : Kubernetes.
- **Interface graphique** : Grafana



4. Composants de l'Application

4.1 - Scrapping / Mocking

- Récupération des données de Wikipédia via son API publique et ingestion dans Kafka.
- On récupère un objet “changement” qui ressemble à ça:


```
{
  "$schema": "/mediawiki/recentchange/1.0.0",
  "meta": {
    "uri": "https://en.wikipedia.org/wiki/Category:Pages_using_WikiProject_banner_shell_with_unknown_parameters",
    "request_id": "2ba93d4f-192b-4983-bb2e-8241624b6fa5",
    "id": "526073d6-67f8-46a6-a94b-8f3871b1a92f",
    "dt": "2025-01-30T10:29:05Z",
    "domain": "en.wikipedia.org",
    "stream": "mediawiki.recentchange",
    "topic": "codfw.mediawiki.recentchange",
    "partition": 0,
    "offset": 1439681113
  },
  "id": 1871006000,
  "type": "categorize",
  "namespace": 14,
  "title": "Category:Pages using WikiProject banner shell with unknown parameters",
  "title_url": "https://en.wikipedia.org/wiki/Category:Pages_using_WikiProject_banner_shell_with_unknown_parameters",
  "comment": "[[:Talk:Yevgen Sotnikov]] removed from category",
  "timestamp": 1738232945,
  "user": "Cewbot",
  "bot": true,
  "notify_url": "https://en.wikipedia.org/w/index.php?diff=1272831437&oldid=1218224312",
  "server_url": "https://en.wikipedia.org/",
  "server_name": "en.wikipedia.org",
  "server_script_path": "/w",
  "wiki": "enwiki",
  "parsedcomment": "<a href="/wiki/Talk:Yevgen_Sotnikov" title="Talk:Yevgen Sotnikov">Talk:Yevgen Sotnikov</a> removed from category"
}
```


4.1 - Kafka

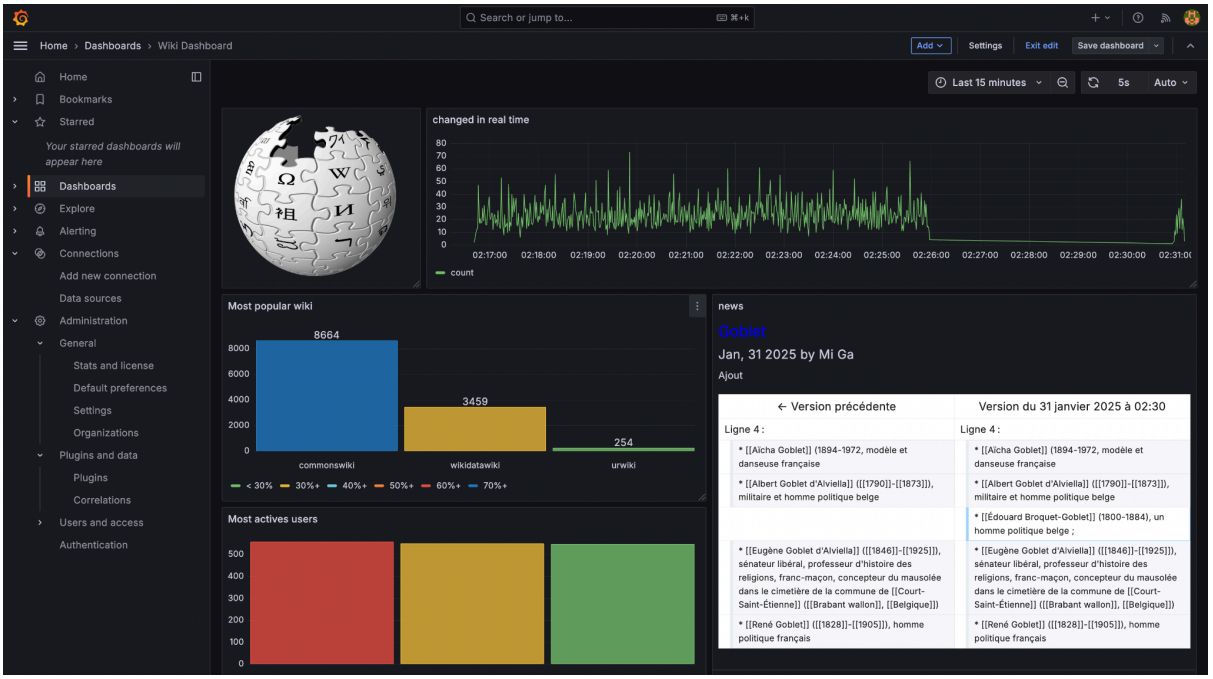
- Utilisation de Kafka comme bus de message pour assurer la scalabilité et la rapidité du traitement des données.

- Implémentation des producers pour la collecte et des consumers pour le traitement et le stockage.
- déploiement avec helm

4.1 - interface Graphique

Développement d'une interface graphique directement sur Grafana permettant de visualiser les tendances et anomalies détectées sur Wikipédia.

Nous avons pu suivre notre maquette et la mettre en place comme vu sur ce screenshot.



Comme dit précédemment, on a adapté certains visuels, et ajouté des flux.
On a pu ajouter une flux de prévisualisation des changements des articles en français, et mis les changements sur une courbe temporelle.

5. Traitement de la donnée

5.1 - Filtrage et Traitement

Pour avoir la donnée souhaitée en résultat, la donnée de base est déjà bien formatée.
Ce que nous faisons est de filtrer des champs inutiles à nos fins, pour avoir une donnée plus affinée et moins volumineuse, vu la quantité que nous stockons et traitons.

5.2 - Stockage

Nous stockons ensuite nos données sur Clickhouse, pour une implémentation simple avec grafana.

6. Problèmes rencontrés et Solutions

6.1 - Problème d'OS

Mac ne prend pas en charge la solution Kafka Streams, rendant 2 de nos développeurs dans l'incapacité de travailler sur cette partie de la solution.

On a donc dû répartir le travail en tenant en compte de ce paramètre, et tester la solution complète sur des PC Linux.

6.1 - Problèmes de données

Plusieurs fois nous avons dû changer de sujet et d'API, dû au fait de la difficulté d'accéder à une API stream accessible et gratuite, et le manque d'intérêt du projet.

7. Conclusions

7.1 - Par rapport aux Objectifs

Nos objectifs ont été remplis, avec l'infrastructure mise en place, les données récoltées, et les dashboards accessibles.

7.2 - Possibles améliorations

Comme dit dans la problématique, lier les données actuelles avec le trafic et l'intérêt d'un article aurait pu améliorer l'analyse de risque et donner une vision plus claire.

7.3 - Notions Vues

Au cours de ce projet, nous avons eu l'occasion de:

- découvrir et déployer Kafka sur un cluster kubernetes
- scrapper une API streaming en temp réel
- réaliser des maquettes d'interface graphique
- créer toute une pipeline de récupération et mise à disposition de données

8. References & Appendices

- GitHub Repository: https://github.com/Hy0g0/Crypto-Data
- Stanford Blog post: https://fsi.stanford.edu/news/wikipedia-part-two