

PROJET TRAFFIC RENNES

Équipe
Zaina TOIBIBOU
Khaly NIANG
Frédéric DUARTE

07/05/2021

Sommaire

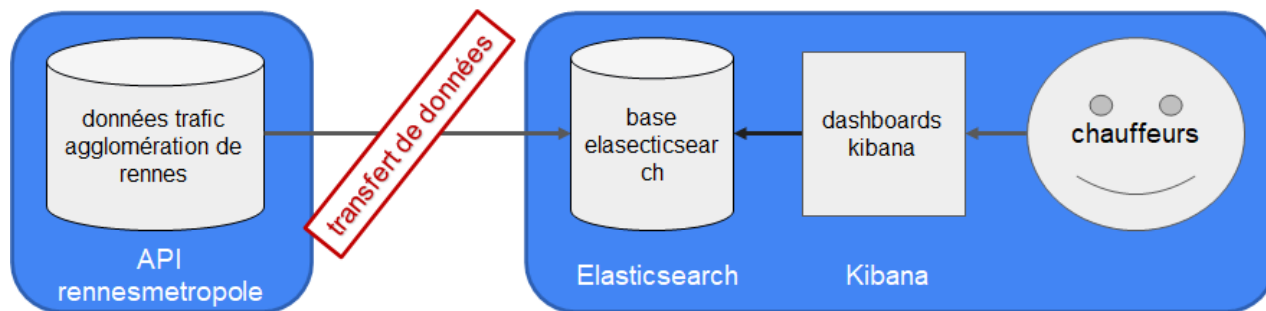
- Introduction
- Architecture de la solution finale
- Etape 1 : le transfert des données
- Etape 2 : la réalisation de dashboards
- Démo
- Amélioration
- Méthodologie de travail

Introduction

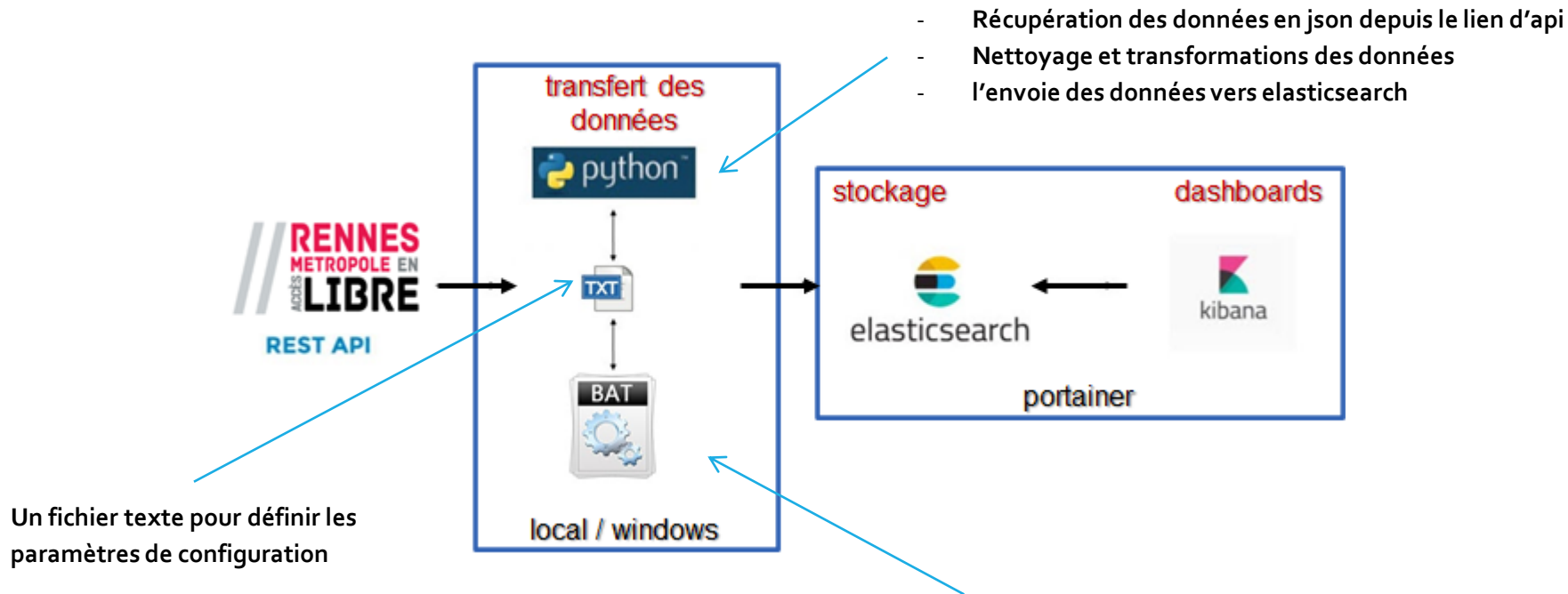
- Client : Transport Rennes
- Besoin : visualiser le trafic en temps réel de l'agglomération rennaise
- Limites :
 - Source de données : api rennesmetropole
 - Màj toutes les 3 minutes, 1000 exportations par appel, niveau de confiance $\geq 50\%$
 - Utilisation de la suite ELK + python

Introduction

Représentation applicative de la solution



Architecture de la solution finale



Le transfert des données (1) : fonctionnement

- Outils : python (+ batch en complément)
- 3 sous-étapes :
 1. appel à l'api et récupération des données en json
 2. nettoyage de ces données
 - filtrage de 1er niveau pour ne garder que le contenu du champ « records »
 - ce champ contient les données de trafics
 - filtrage de 2nd niveau pour « traveltimereliability >= 50 »
 - pour la fiabilité des données
 3. injection de ces données dans elasticsearch :
 - connection à elasticsearch
 - définition du mapping adéquat pour les champs géospatiales
 - fields.geo_point_2d, fields.geo_shape (routes), geometry.coordinates
 - création d'un index vide sous elasticsearch
 - export des données vers elasticsearch

FLUX - CONTINU

Le transfert des données (2) : paramétrisation

- Le fichier des paramètres : *trafficroennes_transfertdata_parameters.txt*

Paramètre	Définition	Type	Par défaut
index_name	nom de l'index	string	traffic_rennes
index_init	créer ou mettre-à-jour l'index	boolean	False
traffic_nb_rows	nombre de ligne par requête api	integer	1000
traffic_reliability	niveau de confiance des données en %	integer	50
traffic_time_interval	durée d'attente entre chaque flux en s	integer / eval	60*3
traffic_time_max	durée total du flux-continue en s	integer / eval	60*60*2

Le transfert des données (3) : paramétrisation

- Le fichier de batch : *.trafficrennes_transfertdata_run.bat*

Paramètre	Définition	Type	Par défaut
python_exe	chemin vers l'exécutable de python		python
python_script	chemin vers le programme python		trafficrennes_transfertdata.py
params_file	chemin vers le fichier des paramèt		trafficrennes_transfertdata_parameters.txt

- Autres fichiers :
 - *trafficrennes_transfertdata.py* : programme python
 - *trafficrennes_transfertdata_utils.py* : code python avec les fonctions personnalisés

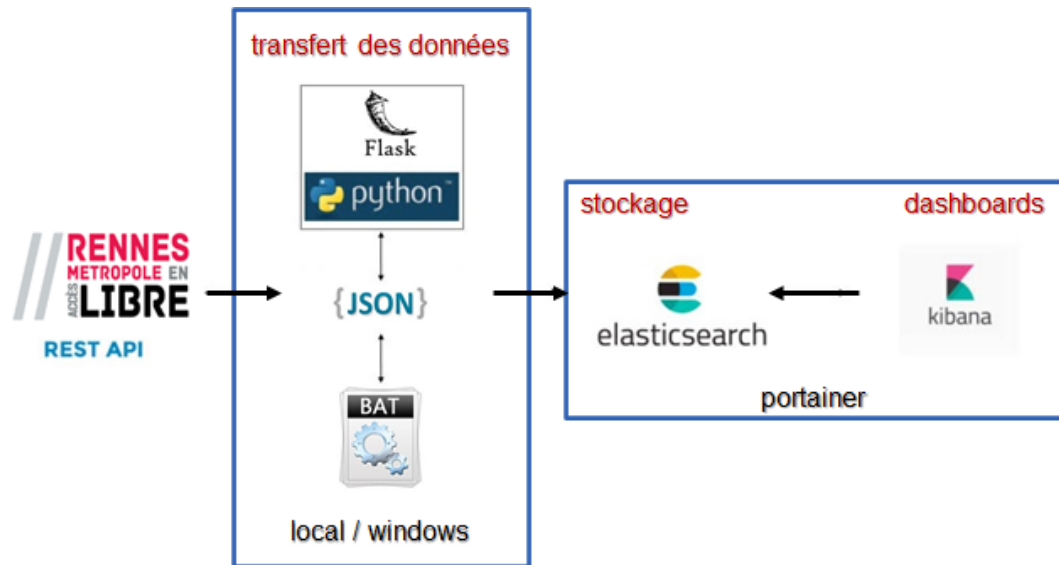
Le transfert des données (4) : suivi et test

- Log affiché dans le terminal, avec :
 - Informations sur les paramètres
 - Heure de début/fin de chaque étape importantes
- Tests :
 - Si l'api répond
 - Si le serveur elastic est actif
 - Si tous les fichiers nécessaires existent
 - Si le type des paramètres de configuration sont ok et si leurs valeurs sont cohérentes
 - → *unittest-v3.py* : programme python qui réalise les test
 - Traitement d'erreurs du programme principale :
 - Si le serveur elastic n'est pas actif → arrêt
 - Si l'api ne répond pas → on retourne un json vide (et on passe au flux suivant)

Démo

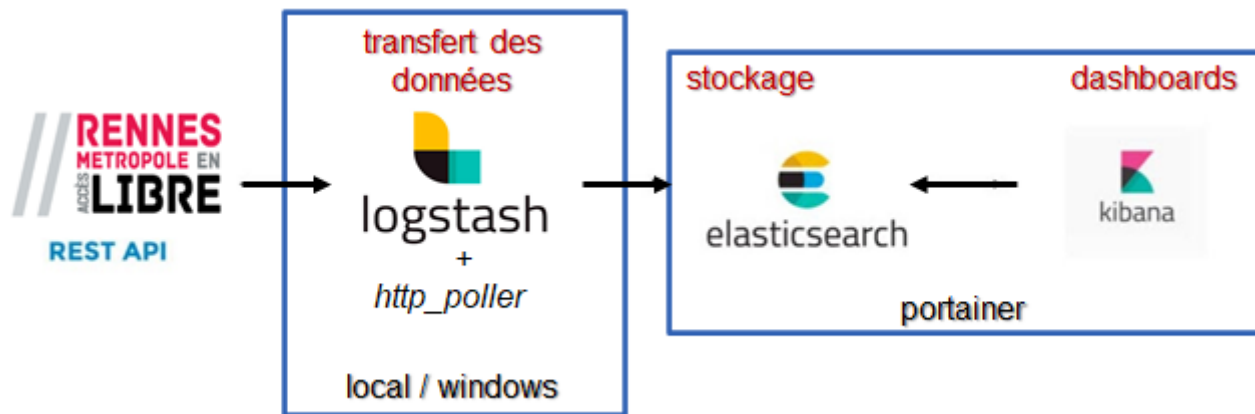
- Prérequis :
 - server elasticsearch et kibana actifs
 - Python3

Amélioration (1) : Evolution de la Solution



1. Réalisation d'un formulaire plus intuitif que l'édition d'un fichier texte
2. Les données sont sauvegardées au format json
3. l'avantage de JSON est qu'il est utilisé pour sérialiser et transmettre des données structurées.

Amélioration (2) : Solution abandonnée



1. Fichier de configuration pour démarrer Logstash, de créer un pipeline et définir les paramètres
2. Le plugin HTTP Poller permet d'appeler une HTTP API, de décoder la sortie de celle-ci en événements et de les envoyer à ElasticSearch
3. Le mappage des données ne nous a pas donné un résultat prévisionnel.

Installation ? Guide ? ou Demo

1. Libraire Elastisearch pour python
2. Scheduler de windows
3. lancement automatique du fichier batch
4. Ouvrir docker , port 9000
5. Accès au dashboard de kibana port 5600

- Agile :
 - Kanban
- DevOps :
 - Git

Traffic_Rennes kanban : 🔍 Search & Filter ⌵ 11 9

En cours 1/3 + Terminé 10 + Pushé 6 + 1

A réaliser Abandon

High
Doc : rédigeons

zaina3-dev add repertoire documents et binaires 3ed8891 9 hours ago 59 commits

binaires	add repertoire documents et binaires	9 hours ago
documents	add repertoire documents et binaires	9 hours ago
releases	maj repertoires	9 hours ago
.gitignore	add repertoire docs	9 hours ago
README.md	add repertoire documents et binaires	9 hours ago
elk_tr.code-workspace		11 days ago

☰ README.md ✎

Projet traffic_rennes

Groupe : Fred, Khaly, Zaina

Sujet

Afficher le trafic en temps réel de la métropole rennaise.

A faire :

- récupérer les données depuis l'api,
- les transférer vers elasticsearch,
- créer le dashboard sur kibana.

API données : <https://data.rennesmetropole.fr/explore/dataset/etat-du-traffic-en-temps-reel/information/>

Utilisation

Prérequis :

Fin

- Des questions ?