Analyse de logs en temps-réel pour `Chaussettes.io`

Sylvain Gault

30 mai 2025

1 Introduction

Ce projet est à réaliser par trois ou quatre en Python sous l'environnement de développement de votre choix. Des livrables seront à rendre et vous en ferez une présentation à une date convenue ultérieurement.

2 Cahier des charges

2.1 Contexte

Vous êtes embauché par l'entreprise `Chaussettes.io`, spécialisée dans la vente en ligne de chaussettes personnalisées à base de laine d'alpaga, a récemment migré ses services vers le cloud. Elle dispose désormais d'un cluster Kafka pour centraliser tous les événements, dont les HTTP de ses microservices, publiés dans un topic appelé `http-logs`.

La direction technique souhaite mettre en place un système d'analyse automatique de ces logs pour détecter rapidement des comportements anormaux (pics d'erreurs, saturation API, attaques DDoS...) et générer des alertes dans leur système de monitoring.

Le système devra utiliser **Apache Spark**, tout en respectant les exigences de **sécurité** imposées par le service conformité (authentification, chiffrement).

2.2 Description des fonctionnalités

2.2.1 Génération de logs

Bien que ce script ne sera pas déployé en production, il sera indispensable à la validation de votre système. Ce programme ne sera utilisé qu'en interne, par vous et par l'équipe chargée des tests de performance.

Ce script devra générer des lignes de logs dans un format similaire à ceux produits par Apache. Dans sa version avancée, les lignes seront insérées dans un topic Kafka, dans sa version simple, il ne fera que les afficher. Il devra être configurable (sans modifier le code) pour générer des erreurs (404, 401, 501, etc.) à différentes fréquences. La fréquence de génération de logs (nombre de ligne par seconde) devra être également configurable.

2.2.2 Analyse de logs avec Spark Streaming

Ce programme devra lire les lignes de logs depuis un stream et calculer plusieurs métriques :

— Fréquence d'erreurs globale
— Fréquence d'erreurs pour une URL donnée
— Fréquence d'erreurs pour un utilisateur donné

Si ces métriques dépassent un certain seuil défini dans le programme, un événement est généré dans un autre stream.

Dans sa version finale, ce programme devra lire les données depuis un topic kafka `http-logs` et envoyer les alertes dans un topic `alerts`.

2.2.3 Déterminer les seuils

Ce programme Spark batch servira à analyser les logs disponibles. Il devra produire des seuils appropriés pour l'analyse de logs en temps réel. Pour ce faire, il devra calculer les différentes métriques et faire en sorte que les alertes ne soient déclenché que 1% du temps.

2.2.4 Sécurité

La sécurité est bien entendu primordiale en production. Il faudra configurer l'authentification et le chiffrement des communication et du stockage. Ou justifier l'absence de celui-ci.

2.3 Planning et livrables

Votre entreprise vous a planifié un *meeting* le lundi 23/06/2025 à 8h00 CEST dans lequel vous devrez présenter votre avancement. Vos livrables intermédiaires devront être mis sur sur Teams dans le devoir *« Chaussettes.io - Livrables intermédiaires »*, un rendu par groupe à cette date. Ces doivent inclure :

— Le programme de génération de logs
— Le programme spark streaming évaluant au moins une des métriques et générant les alertes
— La documentation de l'état actuel de l'avancement du projet
— La documentation des deux programmes

3 Tâches

Après discussion avec votre chef de projet, voici la liste des tâches dans l'ordre approximatif dans lequel vous devriez les réaliser.

3.1 Version minimale

Le programme de génération de logs dans sa version minimale n'est presque pas configurable. Il génère une ligne par seconde, les IP source, URL, code HTTP sont aléatoires. Dans cette version, le programme affiche les lignes avec `print`.

L'analyse de logs minimale lit les logs depuis une socket et n'interagit pas avec Kafka. Les résultats sont stockés dans un fichier.

3.2 Version plus complète

Le programme de génération de logs devra être configurable via des arguments sur la ligne de commande. (En utilisant `argparse` par exemple.) Un certain pourcentage (configurable) des utilisateurs devra générer un certain pourcentage (configurable) d'erreurs sur un certain pourcentage (configurable) d'URLs. La vitesse de génération devra aussi être configurable.

Le chiffrement et l'authentification devront aussi être configurés sur votre déploiement Spark afin d'être prêt à être mis en production.

3.3 Version finale

Kafka devra être intégré dans la vesion finale. Le programme de génération de logs devra utiliser le module `kafka-python` pour écrire dans un topic kafka. Le programme d'analyse de logs devra lire les événement depuis Kafka.

3.4 L'automatisation du déploiement et la documentation

Vous allez devoir collaborer avec d'autres équipes qui réaliseront des tests de performance et de scalabilité. Automatisez autant que faire se peut le déploiement de votre infra de test et de prod.

Documentez comment votre infra est construite et comment interagir avec elle. Comment un nouveau dev rajoute une métrique ? Ou un nouveau job Spark Streaming ?

3.5 Améliorations possibles

Si vous avez du temps, implémentez la génération de logs de manière plus réaliste. Implémentez aussi des méthodes de détection plus intelligentes. Cela demandera probablement d'analyser des vrais logs. Documentez vos découvertes.