État d’avancement du projet — Chaussettes.io

Date : 23 juin 2025

Projet : Analyse de logs en temps réel avec Spark & Kafka

Groupe : Carlos CEREN

✅ Objectifs réalisés à ce jour

1. Générateur de logs (`log_gen.py`) — ✅ Complet

- Génère des logs réalistes au format Apache Combined Log.
- Configurable via la ligne de commande (`argparse`) :
  - Taux d’erreur global
  - Taux d’erreur par IP
  - Taux d’erreur par URL
  - Vitesse de génération
- Envoi des logs dans Kafka (`http-logs`) via `kafka-python` .
- Mode alternatif console pour debug ou test hors Kafka.
- Simulation de comportements réalistes :
  - Utilisateurs problématiques
  - Endpoints défaillants

2. Analyse Spark Streaming (`sparkStreaming.py`) — ✅ Complet

- Lit les logs en temps réel depuis Kafka (`http-logs`).
- Calcule les métriques :
  - Taux d’erreur global
  - Taux d’erreur par IP
  - Taux d’erreur par URL

- Compare les métriques aux seuils (`seuils.json`).
- Envoie les alertes dans Kafka (`alerts`) si dépassement de seuil.
- Détection batchée toutes les 10 secondes avec `foreachBatch`.

3. Calcul des seuils (`sparkSeuilCalcul.py`) — ✅ Complet

- Job Spark batch lisant les logs historiques de Kafka.
- Calcule automatiquement les seuils dynamiques :
  - Seuil global basé sur un pourcentage de taux d'erreur global
  - Seuils IP et URL basés sur les 99e percentiles
- Génère un fichier `seuils.json` utilisé par le streaming.
- Valide les seuils avec une simulation pour garantir ~1% d'alertes.

4. Infrastructure & automatisation — ✅ Opérationnelle

- **Docker Compose** complet :
  - Kafka, Zookeeper, Spark Master, Spark Worker
  - Générateur de logs, job Spark Streaming, job Spark Batch
- **Makefile** :
  - Démarrage / arrêt des services
  - Logs ciblés
  - Exécution du batch avec profil
  - Commandes Kafka (topics, consumers)
  - Nettoyage volumes (`make clean-volumes`)
- Dossiers montés pour persistance des données (`./data`, `./app`)
- Support des profils Docker pour lancer uniquement certaines tâches (`-profile batch`).

⚠️ Points restants à valider / améliorer

- **Sécurité :** Le chiffrement et l'authentification Spark/Kafka ne sont pas encore configurés.
  - Justification en cours (non requis en environnement de test).

- **Visualisation / monitoring :** Pas encore de tableau de bord pour afficher les alertes.
    - Peut être ajouté avec Grafana + Prometheus + connecteur Kafka.

📁 Livrables fournis

- `log_gen.py` — Générateur configurable de logs
- `sparkStreaming.py` — Analyse Spark Streaming
- `sparkSeuilCalcul.py` — Job Spark Batch pour calcul des seuils
- `docker-compose.yml` — Déploiement complet
- `Makefile` — Interface d'automatisation des tâches
- `seuils.json` — Seuils dynamiques générés
- Documentation technique par composant (générateur, streaming)
