Rapport TP3 : Analyse de mots

DZIRI Samy

Étape 1 – « Vive les petits producteurs »

- J'ai commencé par cloner le dépôt GitHub du producteur de messages.
- Ensuite, j'ai étudié et modifié le code, notamment dans le fichier server.js, pour intégrer correctement la configuration de la variable `NUMBER_WORD`.
- J'ai ajusté le fichier .env pour définir les paramètres de génération des messages et m'assurer que le producteur se connecte bien au broker RedPanda.
- J'ai construit l'image Docker du producteur et lancé le conteneur.
- J'ai observé les logs du conteneur pour m'assurer que les messages étaient bien envoyés au broker RedPanda et j'ai résolu quelques problèmes de connexion au réseau Docker.

J'ai rencontré plusieurs difficultés techniques. Tout d'abord, j'ai eu des problèmes de connexion au broker RedPanda, avec des erreurs `ECONNREFUSED`, ce qui a nécessité des ajustements dans la configuration des adresses et des ports dans le fichier **.env** ainsi que dans le fichier de configuration du conteneur. Ensuite, des soucis liés à la configuration du réseau Docker ont émergé, notamment des erreurs de résolution d'adresses comme `redpanda-0`, ce qui m'a poussé à vérifier et modifier les paramètres dans le fichier **docker-compose.yml**. Par ailleurs, le temps de démarrage des services a été un peu long, notamment pour le cluster RedPanda, ce qui m'a obligé à redémarrer plusieurs fois les services afin de vérifier que tous les ports étaient correctement ouverts. Enfin, bien que le mode debug ait été activé, j'ai dû résoudre des avertissements concernant la partition du producteur Kafka avec la version 2.0.0, ce qui m'a conduit à ajuster la configuration du producteur pour supprimer ces messages. Ces difficultés ont été surmontées grâce à une révision minutieuse des fichiers de configuration et des redémarrages des services pour assurer la stabilité du système.

```
  topic: 'mon-super-topic',
  user: 'Floki',
  message: 'incididunt eu est'
}
{
  topic: 'mon-super-topic',
  user: 'Thor',
  message: 'tempor pariatur incididunt'
}
{
  topic: 'mon-super-topic',
  user: 'Thor',
  message: 'nostrud nostrud aute'
}
{ topic: 'mon-super-topic', user: 'Floki', message: 'sit anim dolore' }
```

```
2025-02-04 11:57:26   topic: 'mon-super-topic',
2025-02-04 11:57:26   user: 'Viggo',
2025-02-04 11:57:26   message: 'non fugiat reprehenderit'
2025-02-04 11:57:26 }
2025-02-04 11:57:27 {
2025-02-04 11:57:27   topic: 'mon-super-topic',
2025-02-04 11:57:27   user: 'Frigg',
2025-02-04 11:57:27   message: 'adipisicing elit adipisicing'
2025-02-04 11:57:27 }
2025-02-04 11:57:28 {
2025-02-04 11:57:28   topic: 'mon-super-topic',
2025-02-04 11:57:28   user: 'Harald',
2025-02-04 11:57:28   message: 'ea voluptate voluptate'
2025-02-04 11:57:28 }
2025-02-04 11:57:29 {
2025-02-04 11:57:29   topic: 'mon-super-topic',
2025-02-04 11:57:29   user: 'Frigg',
2025-02-04 11:57:29   message: 'nulla minim culpa'
2025-02-04 11:57:29 }
2025-02-04 11:57:30 {
2025-02-04 11:57:30   topic: 'mon-super-topic',
2025-02-04 11:57:30   user: 'Thora',
2025-02-04 11:57:30   message: 'exercitation esse non'
2025-02-04 11:57:30 }
2025-02-04 11:57:31 {
2025-02-04 11:57:31   topic: 'mon-super-topic',
2025-02-04 11:57:31   user: 'Liv',
2025-02-04 11:57:31   message: 'deserunt exercitation ea'
2025-02-04 11:57:31 }
2025-02-04 11:57:32 {
2025-02-04 11:57:32   topic: 'mon-super-topic',
2025-02-04 11:57:32   user: 'Thora',
2025-02-04 11:57:32   message: 'aliquip adipisicing nulla'
2025-02-04 11:57:32 }
```

Étape 2 – Star de Cinéma : « 26l / 100 km » - Qui suis-je ?

Dans l'étape 2, j'ai commencé par créer un fichier `consumer.js` dans le même projet que le producteur. Ensuite, je me suis abonné au topic `mon-super-topic` en utilisant la bibliothèque KafkaJS. Après m'être connecté au broker, j'ai écrit une fonction pour consommer les messages envoyés vers ce topic. Pour chaque message reçu, j'ai affiché son contenu dans la console. De plus, j'ai créé une fonction pour formater le timestamp de chaque message, afin de l'afficher dans un format lisible (`dd/mm/yyyy à hh:mm`). Cela m'a permis de valider que les messages étaient correctement consommés et traités.
La difficulté principale a été la gestion de la connexion au broker RedPanda. Il fallait s'assurer que l'URL et le port étaient correctement définis pour l'abonnement au topic. Une autre difficulté était le formatage du timestamp de Kafka, qui devait être converti en un format lisible.

Étape 3 –« Are you Redis »?

J'ai modifié le projet consommateur pour découper les messages en mots. Ensuite, j'ai intégré Redis en utilisant la bibliothèque `redis` pour Node.js afin d'incrémenter les compteurs des mots. Cependant, je n'ai pas réussi à tout finir. Bien que j'ai réussi à faire fonctionner la découpe des messages et à intégrer Redis, je n'ai pas pu faire fonctionner l'affichage en temps réel des mots sur le site web. Il me reste encore quelques ajustements à faire pour que l'ensemble des services communiquent correctement et que l'affichage soit mis à jour.

Les difficultés rencontrées concernaient principalement la configuration et la connexion au serveur Redis. Il était essentiel de bien s'assurer que Redis était correctement configuré et que le consommateur pouvait interagir avec lui sans problème.