

Compte rendu

Développement Avancé : TP 3

Laurent Giustignano

Par Steven TEA 303

Étape 1

Pour la première étape du TP, j'ai dû la réaliser deux fois, chacune de manière différente. N'ayant pas d'ordinateur à disposition à l'IUT, j'ai utilisé un ordinateur de l'établissement avec une machine virtuelle pour lancer Docker. Pour ce faire, j'ai effectué des modifications dans le programme fourni afin de permettre cette configuration. Il a été nécessaire d'ajouter l'adresse IP de la machine virtuelle dans le fichier docker-compose.yml, ce qui a entraîné des modifications des brokers dans les listes de producteurs et d'administration, afin de les connecter au port externe du serveur Kafka.

Étape 2

Pour développer les consommateurs, j'ai créé un nouveau projet dédié, contenant uniquement les consommateurs du serveur RedPanda. J'ai repris le fichier de configuration du projet initial afin de pouvoir me connecter au serveur RedPanda et m'abonner au topic "mon-super-topic" pour lire les messages envoyés.

J'ai rencontré quelques difficultés lors du développement de la fonction connexion(). Par exemple, je n'avais pas pris en compte le caractère asynchrone des opérations de connexion et d'abonnement aux topics du consommateur. J'ai réalisé qu'il était nécessaire d'utiliser await pour m'assurer que la connexion s'effectue systématiquement avant l'abonnement.

De plus, lors des tests de récupération de messages, j'ai réalisé que chaque message récupéré par un consommateur ne serait pas disponible lors de l'exécution ultérieure du programme. Par conséquent, au lieu de relancer le programme de producteur et de consommateur simultanément pour récupérer de nouveaux messages, j'ai essayé de modifier l'identifiant du groupe pour accéder aux anciens messages. Cela a entraîné une complication inutile dans le processus de test.

Étape 3

Une fois le projet React récupéré, j'ai continué à travailler sur le projet cons-red-panda, pour que lorsqu'un message est reçu par le consommateur, il va incrémenter la valeur du nombre d'occurrence de chaque mot.

Occurrence Des Mots	
Nom du mot	Nombre d'occurrence
deserunt	14
cillum	13
commodo	11
culpa	11
proident	11
in	11
mollit	10
laborum	9
minim	8
aliqua	8
anim	8
qui	6
sit	6
incididunt	5

Pour cela, je me connecte sur le serveur Redis, avec en tant que host “myredis” et en port 6379. Puis, pour chaque message, je sépare les mots par un espace et je les compte indépendamment.

La grande difficulté de cette étape a été de me connecter avec le serveur Redis, ne sachant pas quoi mettre pour le numéro de port et en adresse du serveur, je n’arriverais pas à y accéder. Donc, j’ai essayé de créer une image Docker avec le consommateur pour essayer de m’y connecter en utilisant --network au conteneur occurrence-mots, cependant, en faisant cela, je perdais la connexion entre le consommateur et le serveur Redpanda. J’ai finalement trouvé de quelle manière je peux me connecter au serveur Redis, en regardant dans le fichier docker-compose.yml du projet occurrence-mots, j’ai pu trouver le lien, donc le nom du conteneur de redis, “myredis” et le port exposé est 6379.