DATA ENGINEER

INPT
المعهد الوطني للبريد والمواصلات
Institut National des Postes et Télécommunications

Rapport

Maquette pour l'enregistrement des messages Kafka dans une base de données NoSQL

Réalisé par :

. **Abdellah AGHLALOU**

. **Kamal ADDI**

. **Abderrahim ALAKOUCHE**

DATA Engineer INE3

Code Github :
https://github.com/addi-kamal/Enregistrement-des-messages-Kafka-dans-MongoDB

I. Installer Apache Kafka sur Ubuntu 20.04 :

1. Installation de Java:

Pour configurer Kafka sur Ubuntu, il faut d'abord installer Java.

```
$ sudo apt update
$ sudo apt install default-jdk
```

2. Télécharger la dernière version d'Apache Kafka:

```
$ wget http://www-us.apache.org/dist/kafka/2.7.0/kafka_2.13-2.7.0.tgz
$ tar xzf kafka_2.13-2.7.0.tgz
$ mv kafka_2.13-2.7.0 /usr/local/kafka
```

3. Création de fichiers d'unité Systemd:

```
$ vim /etc/systemd/system/zookeeper.service
```

Après la création de fichier d'unité Systemd pour **zookeeper**, il faut ajouter le contenu suivant:

```
[Unit]
Description=Apache Zookeeper server
Documentation=http://zookeeper.apache.org
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
ExecStart=/usr/local/kafka/bin/zookeeper-server-start.sh
/usr/local/kafka/config/zookeeper.properties
ExecStop=/usr/local/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
```

Ensuite, création d'un fichier d'unité systemd pour le service Kafka :

```
$ vim /etc/systemd/system/kafka.service
```

Après la création de fichier d'unité Systemd pour **kafka**, il faut ajouter le contenu suivant:

```
[Unit]
Description=Apache Kafka Server
Documentation=http://kafka.apache.org/documentation.html
Requires=zookeeper.service

[Service]
Type=simple
Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
ExecStart=/usr/local/kafka/bin/kafka-server-start.sh
/usr/local/kafka/config/server.properties
ExecStop=/usr/local/kafka/bin/kafka-server-stop.sh

[Install]
WantedBy=multi-user.target
```

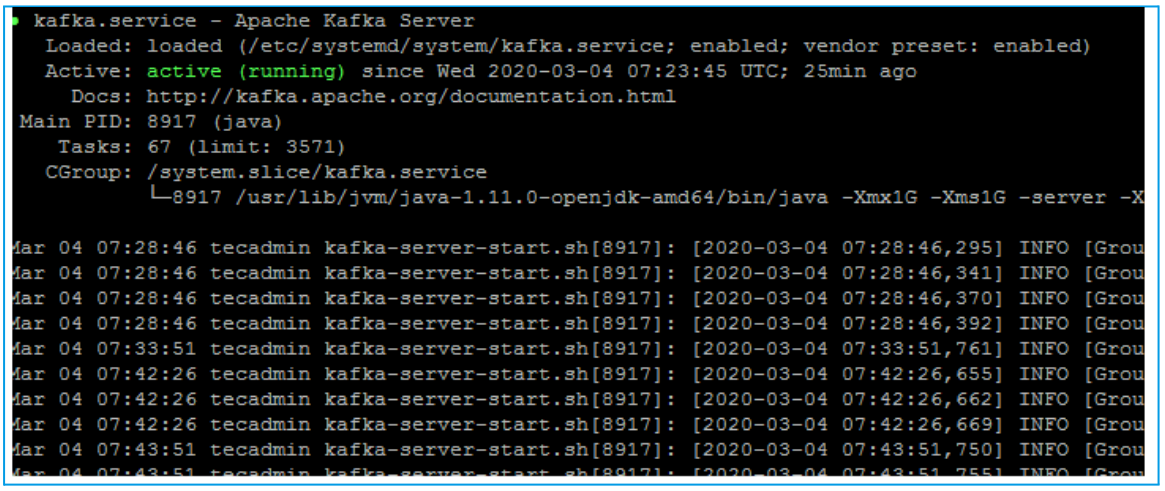
Rechargement du systemd daemon pour appliquer les nouvelles modifications.

```
$ systemctl daemon-reload
```

4. **Démarrer le service Kafka et Zookeeper:**

Nous devons démarrer le service ZooKeeper, puis démarrer Kafka.

```
$ sudo systemctl start zookeeper
$ sudo systemctl start kafka
$ sudo systemctl status kafka
```



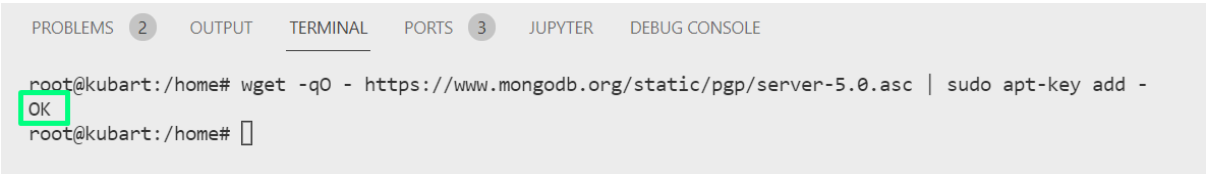
II. Installer MongoDB sur le serveur Ubuntu :

https://addi-kamal.medium.com/mongodb-iceberg-how-to-install-mongodb-on-ubuntu-server-9c5beea2c62

Pour installer MongoDB sur Ubuntu, on exécute la commande suivante pour importer la clé GPG publique MongoDB à partir de https://www.mongodb.org/static/pgp/server-5.0.asc :

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

L'opération doit répondre par un OK :



On crée un fichier /etc/apt/sources.list.d/mongodb-org-5.0.list en utilisant la commande suivante :

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0
```

```
multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-5.0.list
```

On exécute la commande suivante pour recharger la base de données de packages locale :

```
sudo apt-get update
```

```
root@kubart:/home# apt-get update
Hit:1 http://mirrors.digitalocean.com/ubuntu focal InRelease
Ign:3 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 InRelease
Hit:4 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu focal-updates InRelease
Hit:6 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 Release
Hit:7 http://mirrors.digitalocean.com/ubuntu focal-backports InRelease
Get:8 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 https://downloads.apache.org/cassandra/debian 40x InRelease
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [13.0 kB]
Fetched 127 kB in 1s (99.7 kB/s)
Reading package lists... Done
```

Installez les packages MongoDB :

Pour installer la dernière version stable de MongoDB, on exécute la commande suivante :

```
sudo apt-get install -y mongodb-org
```

```
(Reading database ... 102757 files and directories currently installed.)
Preparing to unpack .../0-mongodb-org-shell_5.0.5_amd64.deb ...
Unpacking mongodb-org-shell (5.0.5) ...
Selecting previously unselected package mongodb-org-server.
Preparing to unpack .../1-mongodb-org-server_5.0.5_amd64.deb ...
Unpacking mongodb-org-server (5.0.5) ...
Selecting previously unselected package mongodb-org-mongos.
Preparing to unpack .../2-mongodb-org-mongos_5.0.5_amd64.deb ...
Unpacking mongodb-org-mongos (5.0.5) ...
Selecting previously unselected package mongodb-org-database-tools-extra.
Preparing to unpack .../3-mongodb-org-database-tools-extra_5.0.5_amd64.deb ...
Unpacking mongodb-org-database-tools-extra (5.0.5) ...
Selecting previously unselected package mongodb-org-database.
Preparing to unpack .../4-mongodb-org-database_5.0.5_amd64.deb ...
Unpacking mongodb-org-database (5.0.5) ...
Selecting previously unselected package mongodb-org-tools.
Preparing to unpack .../5-mongodb-org-tools_5.0.5_amd64.deb ...
Unpacking mongodb-org-tools (5.0.5) ...
Selecting previously unselected package mongodb-org.
Preparing to unpack .../6-mongodb-org_5.0.5_amd64.deb ...
Unpacking mongodb-org (5.0.5) ...
Setting up mongodb-org-server (5.0.5) ...
Setting up mongodb-org-shell (5.0.5) ...
Setting up mongodb-org-mongos (5.0.5) ...
Setting up mongodb-org-database-tools-extra (5.0.5) ...
Setting up mongodb-org-database (5.0.5) ...
Setting up mongodb-org-tools (5.0.5) ...
Setting up mongodb-org (5.0.5) ...
Processing triggers for man-db (2.9.1-1) ...
root@kubart:/home#
```

MongoDB est maintenant installé avec succès.

```
root@kubart:/home# systemctl status mongod
● mongod.service - MongoDB Database Server
     Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-01-27 22:23:09 UTC; 7s ago
       Docs: https://docs.mongodb.org/manual
   Main PID: 113713 (mongod)
     Memory: 61.6M
     CGroup: /system.slice/mongod.service
             └─113713 /usr/bin/mongod --config /etc/mongod.conf
root@kubart:/home#
```

III. Data Pipeline : Envoyer des messages de Kafka à MongoDB :

1. Producer :

Lecture des messages à partir de Kafka :

Différents types de données peuvent être envoyées par le Producer qui peut être lu par le Consumer.

On va crée dabord un fichier Producer.py avec le script python suivant :

```
# Import KafkaProducer from Kafka library
from kafka import KafkaProducer

# Define server with port
bootstrap_servers = ['localhost:9092']

# Define topic name where the message will publish
topicName = 'My_Topic'

# Initialize producer variable
producer = KafkaProducer(bootstrap_servers =
bootstrap_servers)

while(True):
    print("enter your message :")
    message_to_send = input()
    # Publish text in defined topic
    producer.send(topicName, bytes(message_to_send,
'utf-8'))
```

```
    # Print message
    print("--> Message sent to Consumer")
    print("---------------------------------")
```

Le module ***KafkaProducer*** est importé de la bibliothèque Kafka. La liste des Brokers doit être définie au moment de l'initialisation de l'objet producteur pour se connecter au serveur Kafka. Le port par défaut de Kafka est ***'9092'***.
L'argument **bootstrap_servers** est utilisé pour définir le nom host avec le port.
My_Topic : est défini comme un nom du topic par lequel le message sera envoyé par le producteur. Ensuite, un simple message texte sera récupéré à l'aide de la méthode ***input()*** et sera envoyé à l'aide de la méthode ***send()*** de KafkaProducer, au Consumer.

2. Consumer :

On va crée un autre fichier Cunsomer.py avec le script python suivant :

```
from pymongo import MongoClient
# connect to MongoDB
myclient = MongoClient("mongodb://localhost:27017/")
mydb = myclient["message_db"]
mycol = mydb["message"]

# Import KafkaConsumer from Kafka library
from kafka import KafkaConsumer

# Import sys module
import sys

# Define server with port
bootstrap_servers = ['localhost:9092']

# Define topic name from where the message will recieve
topicName = 'My_Topic'
```

```
# Initialize consumer variable
consumer = KafkaConsumer(topicName, group_id
='group1',bootstrap_servers =bootstrap_servers)

# Read and print message from consumer
for msg in consumer:
    d = {}
    topic_name = msg.topic
    message = msg.value.decode("utf-8")

    d = {"topic_name" : topic_name,
     "message" : message}

    mycol.insert_one(d)

    print("Topic Name=%s"%(topic_name))
    print("Message=%s"%(message))
    print("--> Message sent to MongoDB")
    print("-------------------------------")
# Terminate the script
sys.exit()
```

Le module ***KafkaConsumer*** est importé de la bibliothèque Kafka pour lire les données de Kafka.

Le module sys est utilisé ici pour terminer le script.

Le même nom du host et numéro de port du producteur sont utilisés dans le script du consommateur pour lire les données de Kafka.

Le nom du topic du consommateur et du producteur doit être le même, c'est-à-dire "***My_Topic***". Ensuite, l'objet consommateur est initialisé avec les trois arguments: nom du topic, identifiant de groupe et informations sur le serveur.

La boucle for est utilisée ici pour lire le texte envoyé par le producteur Kafka.

A chaque fois qu'on lit un message, on le stocke dans MongoDB à l'aide de la méthode ***insert().*** Et on affiche un message “--> Message sent to MongoDB”, pour vérifier que notre message a été envoyé à MongoDB.

IV. Demo :

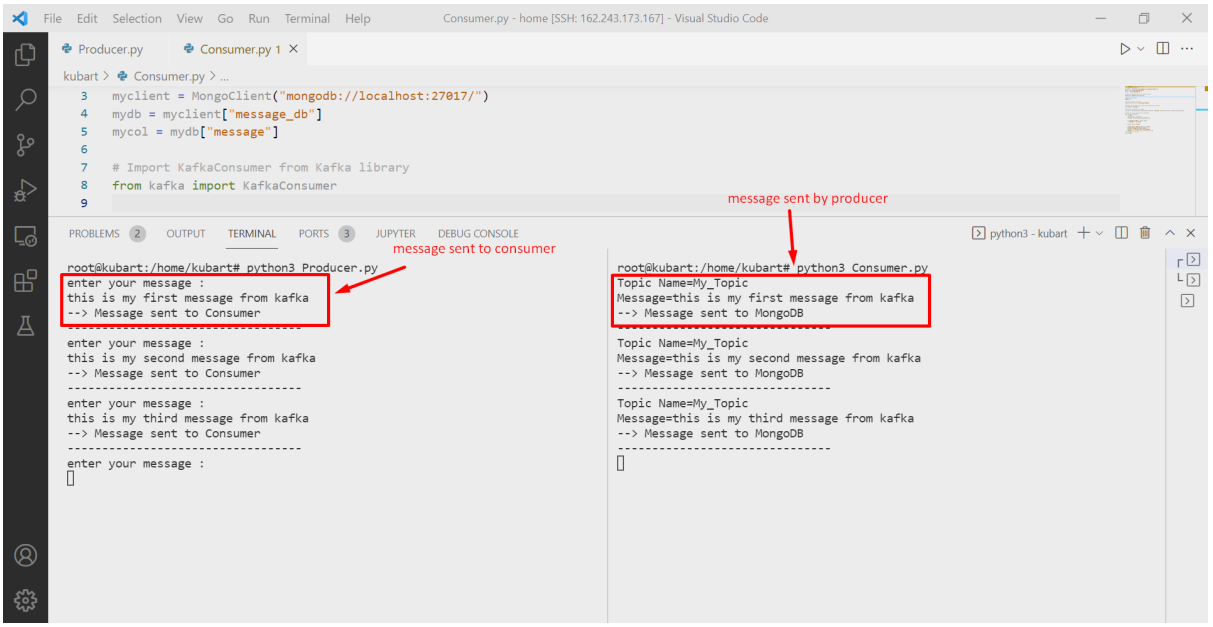
On exécute la commande suivante à partir d'un terminal pour exécuter le script du producteur :

```
$ python3 Producer.py
```

on exécute la commande suivante à partir d'un autre terminal pour exécuter le script consommateur :

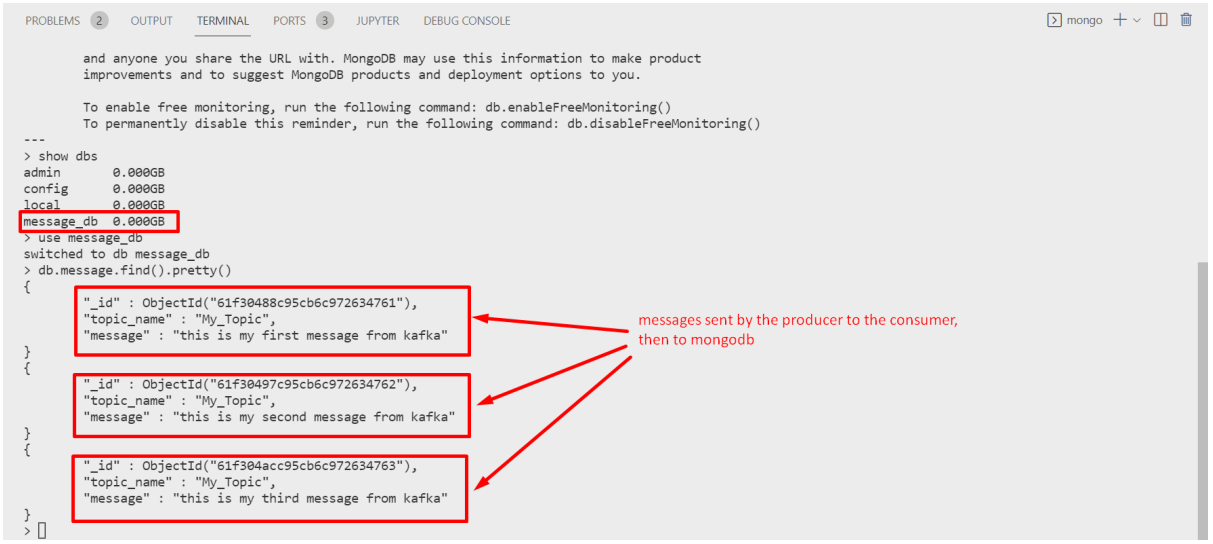
```
$ python3 Consumer.py
```

La sortie affiche le message envoyé et un message : "***--> Message sent to Consumer***" au niveau du Producer. Et le nom du Topic et le message texte envoyé par le producteur au niveau du Consumer.



On verifier sur MongoDB :

On trouve que les messages sont bien stockées dans MongoDB :


PROBLEMS 2
OUTPUT
TERMINAL
PORTS 3
JUPYTER
DEBUG CONSOLE
mongo
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
message_db 0.000GB
> use message_db
switched to db message_db
> db.message.find().pretty()
{
"_id" : ObjectId("61f30488c95cb6c972634761"),
"topic_name" : "My_Topic",
"message" : "this is my first message from kafka"
}
{
"_id" : ObjectId("61f30497c95cb6c972634762"),
"topic_name" : "My_Topic",
"message" : "this is my second message from kafka"
}
{
"_id" : ObjectId("61f304acc95cb6c972634763"),
"topic_name" : "My_Topic",
"message" : "this is my third message from kafka"
}
> messages sent by the producer to the consumer, then to mongodb