EXERCICE : lecture de csv en python et envoi dans un topic un ficher et l’intercepter par un consumer:

1. Envoie de fichier

filepath = xxxx'

producer = KafkaProducer(bootstrap\_servers='localhost:9092')

with open(filepath) as fp:

line = fp.readline()

while line:

producer.send('test', line.encode())

line = fp.readline()

1. Consommation du fichier

from kafka import KafkaConsumer

consumer = KafkaConsumer('test')

for msg in consumer:

print (msg)

Exercice :

Transformer le consumer précèdent en module de transformation. Pour chaque ligne reçue, convertir le contenu selon les règles ci dessous et le transmettre dans la file ‘testalpha’

Dans le cas ou vous utilisez «vins.csv » incrémentez le premier champs de chaque ligne par 1

Et si vous utilisez monument mettez toute la ligne en majuscule .upper()

Exercice :

Pour apprendre à manipuler le KSQL :

étapes du TP:

1) aller dans la racine de confluent-5.0.0

avant de lancer confluent

sudo rm -r /tmp/\*

2) lancer confluent

bin/confluent start

3) charger le module twitter

bin/confluent load twitter\_source -d twitter-source.json

3') vérifier le bon fonctionnement du connecteur

bin/kafka-console-consumer --bootstrap-server localhost:9092 --from-beginning --topic twitter\_source|jq '.Text'

4) créer une table qui collecte les streams et les affiches

CREATE STREAM twitter\_raw (CreatedAt BIGINT, Id BIGINT, Text VARCHAR) WITH (KAFKA\_TOPIC='trump', VALUE\_FORMAT='JSON');

5) charger les offset par ordre temporel croissant

SET 'auto.offset.reset' = 'earliest';

6) afficher le contenu d'un texte chargé

SELECT text FROM twitter\_raw LIMIT 1;

7) supprimer le contenu du texte chargé

DROP stream twitter\_raw;

8) parseur le flux charger en realisant la transformation

CREATE STREAM twitter\_raw (CreatedAt bigint,Id bigint, Text VARCHAR, SOURCE VARCHAR, Truncated VARCHAR, InReplyToStatusId VARCHAR, InReplyToUserId VARCHAR, InReplyToScreenName VARCHAR, GeoLocation VARCHAR, Place VARCHAR, Favorited VARCHAR, Retweeted VARCHAR, FavoriteCount VARCHAR, User VARCHAR, Retweet VARCHAR, Contributors VARCHAR, RetweetCount VARCHAR, RetweetedByMe VARCHAR, CurrentUserRetweetId VARCHAR, PossiblySensitive VARCHAR, Lang VARCHAR, WithheldInCountries VARCHAR, HashtagEntities VARCHAR, UserMentionEntities VARCHAR, MediaEntities VARCHAR, SymbolEntities VARCHAR, URLEntities VARCHAR) WITH (KAFKA\_TOPIC='renault',VALUE\_FORMAT='JSON');

9) charger les twittes dont les tags contiennent china ou tax

SELECT TIMESTAMPTOSTRING(CreatedAt, 'yyyy-MM-dd HH:mm:ss.SSS') AS CreatedAt, EXTRACTJSONFIELD(user,'$.ScreenName') as ScreenName,Text FROM twitter\_raw WHERE LCASE(hashtagentities) LIKE '%gilet%' OR LCASE(hashtagentities) LIKE '%tax%';

10) créer et charger une table stream avec un data cleaning

CREATE STREAM twitter AS SELECT TIMESTAMPTOSTRING(CreatedAt, 'yyyy-MM-dd HH:mm:ss.SSS') AS CreatedAt, EXTRACTJSONFIELD(user,'$.Name') AS user\_Name, EXTRACTJSONFIELD(user,'$.ScreenName') AS user\_ScreenName, EXTRACTJSONFIELD(user,'$.Location') AS user\_Location, EXTRACTJSONFIELD(user,'$.Description') AS user\_Description, Text,hashtagentities,lang FROM twitter\_raw ;

11) extraire de la table twitter les enregistrements dont le text contient tax

SELECT CREATEDAT, USER\_NAME, TEXT FROM TWITTER WHERE TEXT LIKE '%tax%';

12) extraire le nombre de tweet par utilisateur sur l'heure

SELECT user\_screenname, COUNT(\*) FROM twitter WINDOW TUMBLING (SIZE 1 HOUR) GROUP BY user\_screenname HAVING COUNT(\*) > 1;

13) créer une table user\_tweet\_count avec la selection precedente

CREATE TABLE user\_tweet\_count AS SELECT user\_screenname, count(\*) AS tweet\_count FROM twitter WINDOW TUMBLING (SIZE 1 HOUR) GROUP BY user\_screenname ;

14) décrire la table user\_tweet\_count

DESCRIBE user\_tweet\_count;

Exercice stack spark streaming – kafka – cassandra :

1. Lancer le docker

docker run -v `pwd`:/home/guest/host -p 4040:4040 -p 8888:8888 -p 23:22 -ti --privileged yannael/kafka-sparkstreaming-cassandra

1. Lancer le script d’initiation de la base cassandra

sh startup\_script.sh

1. Lancer le notebook

notebook --allow-root

1. Lancer le producer

ouvrir kafkaSendDataPy.ipynb et executer toutes les cellules.

1. Lancer le consumer qui écrit dans cassandra

ouvrir kafkaReceiveAndSaveToCassandraPy.ipynb et executer toutes les cellules pour activer le streaming