



Département Télécommunications, Réseaux & Informatique
Ecole Nationale des Sciences Appliquées d'El Jadida
Université Chouaib Doukkali



Ingénierie Informatique et Technologies Emergentes (IITE)

3eme année Cycle Ingénieur

Architecture Big Data pour le suivi et la surveillance
des événements basés sur les réseaux sociaux avec
Apache Kafka, Apache Atlas, Parquet, Spark Streaming,
Spark NLP, Apache Airflow et Tableau

Réalisé par

Rabab FAHSSI
Houda EL KORAINI

Encadré par

M. Fahd KALLOUBI

Année Universitaire 2023/2024

Table des matières

I. Objectif du projet	4
II. Prérequis pour le projet - Installation et Configuration	4
1. Environnement de Développement	4
2. Configuration de l'Environnement Dockerisé	4
3. Installation et Configuration de Tableau	8
III. Scraping des données avec NtScraper vers un topic Kafka	9
IV. Traitement et Stockage des Données avec Spark et Hive en format Parquet	11
V. Entraînement de modèle avec Spark NLP	13
VI. Visualisation des Résultats avec Tableau	17
VII. Orchestration avec Apache Airflow	23

Liste des figures

Figure 1:fichier docker-compose pour la création des conteneurs docker.....	5
Figure 2:fichier docker-compose pour la création des conteneurs docker.....	6
Figure 3:Exécution des conteneurs	6
Figure 4:Affichage de la liste des conteneurs	7
Figure 5:Liste des conteneurs.....	7
Figure 6:Installation de Tableau Desktop	8
Figure 7:Site de téléchargement de Hive Tableau connector.....	9
Figure 8:Code de Scraping des données avec NtScraper vers un topic Kafka.....	10
Figure 9:les tweets scrapé.....	10
Figure 10:Interface de control center	11
Figure 11:L'ajout des données sur Hive en format parquet	13
Figure 12:Connexion aux cluster Hive.....	18
Figure 13:Etablissement de la connexion entre Hive et Tableau	19
Figure 14>List des Tables	20
Figure 15:Importation des données de Hive sur tableau	20
Figure 16:Le nombre de likes et de retweets reçus par pays pour un tweet.....	21
Figure 17:Le nombre de tweets dans chaque pays	22
Figure 18:Distribution des tweets dans le monde	22
Figure 19:Tableau de bord	23
Figure 20:Connexion à l'interface utiliser.....	24
Figure 21:Interface utilisateur de Airflow	24
Figure 22:Code du dag du projet.....	26
Figure 23:Visualisation de l'exécution du dag créer	27

I. Objectif du projet

L'objectif du projet est de s'intégrer au sein d'une architecture Big Data spécialisée dans la surveillance des événements sur les réseaux sociaux, en mettant l'accent sur Twitter. Cela implique l'exploitation des fonctionnalités d'Apache Airflow pour automatiser le processus de collecte de données à partir de NTScraper, favorisant ainsi une gestion efficace et planifiée des flux d'information provenant de Twitter. Une fois les données collectées, l'objectif est de les visualiser de manière graphique sur Tableau, offrant ainsi une analyse visuelle du nombre de sentiments exprimés sur les réseaux sociaux, segmentés par continent. Cette approche vise à fournir des informations pertinentes et exploitables, facilitant la compréhension des tendances émotionnelles à l'échelle mondiale grâce à une représentation visuelle claire et compréhensible.

II. Prérequis pour le projet - Installation et Configuration

Ce projet nécessite l'installation et la configuration préalable de plusieurs outils et environnements. Suivez attentivement ces étapes pour garantir un déroulement sans accroc du projet.

1. Environnement de Développement

a. Système d'Exploitation :

Vérifiez que vous disposez d'un système d'exploitation compatible avec les outils nécessaires (Exemple : Ici, on utilise Windows).

b. RAM :

Assurez-vous d'avoir une mémoire RAM supérieure à 13 Go.

c. Docker :

- Téléchargez et installez Docker en suivant les instructions spécifiques à votre système d'exploitation

-Vérifiez l'installation avec la commande :

Docker --version

d. Docker Compose :

-Téléchargez et installez Docker Compose en suivant les instructions spécifiques à votre système d'exploitation.

-Vérifiez l'installation avec la commande : docker-compose --version

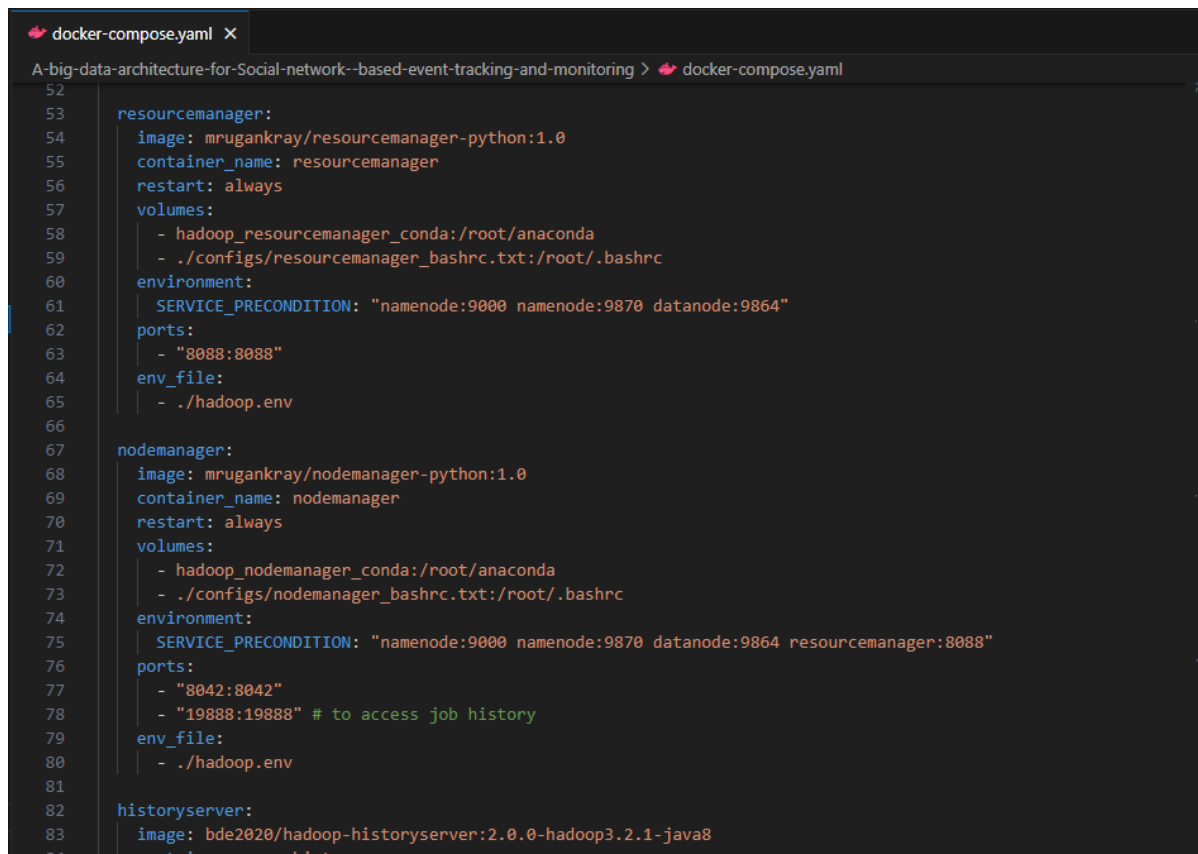
2. Configuration de l'Environnement Dockerisé

a. Création des Conteneurs

Rédigez un fichier docker-compose.yml détaillant les services requis pour cette application

```
docker-compose.yml X
A big-data-architecture-for-Social-network-based-event-tracking-and-monitoring > docker-compose.yml
1  version: "3"
2
3  services:
4    # Setting Up HDFS & YARN #
5    namenode:
6      image: mrugankray/namenode-spark-airflow-flume-zepplin:1.1
7      container_name: namenode
8      restart: always
9      ports:
10       - 9870:9870
11       - 9000:9000
12       - 8082:8082 # zeppelin ui
13       - 8080:8080 # spark master web ui
14       - 8081:8081 # spark slave web ui
15       - 4040:4040 # spark driver web ui
16       - 3000:3000 # airflow ui
17      volumes:
18       - hadoop_namenode:/hadoop/dfs/name
19       - hadoop_namenode_conda:/root/anaconda
20       - hadoop_namenode_spark:/opt/spark
21       - hadoop_namenode_zeppelin:/opt/zeppelin
22       - ./configs/zeppelin-site.xml:/opt/zeppelin/conf/zeppelin-site.xml
23       - ./configs/zeppelin-env.sh:/opt/zeppelin/conf/zeppelin-env.sh
24       - ./configs/namenode_bashrc.txt:/root/.bashrc
25       - ./configs/namenode_airflow.cfg:/root/airflow/airflow.cfg
26       - ./dags:/root/airflow/dags
27       - airflow_namenode:/root/airflow
28       - ./configs/namenode/flume/flume-env.sh:/opt/flume/conf/flume-env.sh
29       - ./flume_config/flume.conf:/opt/flume/conf/flume.conf
30       - hadoop_namenode_flume:/opt/flume
31      environment:
32       - CLUSTER_NAME=hadoop-learning
```

Figure 1: fichier docker-compose pour la création des conteneurs docker



```
52
53 resourcemanager:
54   image: mrugankray/resourcemanager-python:1.0
55   container_name: resourcemanager
56   restart: always
57   volumes:
58     - hadoop_resourcemanager_conda:/root/anaconda
59     - ./configs/resourcemanager_bashrc.txt:/root/.bashrc
60   environment:
61     SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
62   ports:
63     - "8088:8088"
64   env_file:
65     - ./hadoop.env
66
67 nodemanager:
68   image: mrugankray/nodemanager-python:1.0
69   container_name: nodemanager
70   restart: always
71   volumes:
72     - hadoop_nodemanager_conda:/root/anaconda
73     - ./configs/nodemanager_bashrc.txt:/root/.bashrc
74   environment:
75     SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864 resourcemanager:8088"
76   ports:
77     - "8042:8042"
78     - "19888:19888" # to access job history
79   env_file:
80     - ./hadoop.env
81
82 historyserver:
83   image: bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8
84   container_name: historyserver
```

Figure 2:fichier docker-compose pour la création des conteneurs docker

b. Exécution des conteneurs

Pour démarrer les conteneurs définis dans votre fichier docker-compose.yml et les exécuter en arrière-plan, utilisez la commande suivante : docker-compose up -d



```
PS C:\Users\Bell\Desktop\sentiment_analysis_cluster> docker-compose up -d
[+] Running 0/0
 - historyserver Pulling                                0.1s
 - hive-server Pulling                                  0.1s
 - zookeeper Pulling                                    0.1s
 - resourcemanager Pulling                              0.1s
 - control-center Pulling                               0.1s
 - schema-registry Pulling                             0.1s
 - hive-metastore Pulling                               0.1s
```

Figure 3:Exécution des conteneurs

Une fois les conteneurs lancés, vous pouvez lister les identifiants des conteneurs en cours d'exécution et leurs ports associés en utilisant la commande : docker ps

```
C:\Users\Dell\Desktop\sentiment_analysis_cluster>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
86d6777001e1	confluentinc/cp-enterprise-control-center:5.4.0	"/etc/confluent/dock..."	2 days ago	Up 21 minutes	0.0.0.0:9021->9021/tcp
d572dfc53358	confluentinc/cp-schema-registry:5.4.0	"/etc/confluent/dock..."	2 days ago	Up 21 minutes	8081/tcp, 0.0.0.0:8083
0ab6e9e09c16	confluentinc/cp-server:5.4.0	"/etc/confluent/dock..."	2 days ago	Up 21 minutes	0.0.0.0:9092->9092/tcp
c0bd3c0ba446	mrugankray/hive-server-sqoop:1.0	"entrypoint.sh /bin/..."	2 days ago	Up 21 minutes	0.0.0.0:10000->10000/t
baaefabebfe5	confluentinc/cp-zookeeper:5.4.0	"/etc/confluent/dock..."	2 days ago	Up 21 minutes	2888/tcp, 0.0.0.0:2181
2ff62f0880f6	bde2020/hive:2.3.2-postgresql-metastore	"entrypoint.sh /opt/..."	2 days ago	Up 21 minutes	10000/tcp, 0.0.0.0:908
642b59511fdc	mrugankray/namenode-spark-airflow-flume-zepplin:1.1	"entrypoint.sh /sta..."	2 days ago	Up 21 minutes (healthy)	0.0.0.0:3000->3000/tcp
6e0ba6d0ffaf	mrugankray/nodemanager-python:1.0	"entrypoint.sh /run..."	2 days ago	Up 21 minutes (healthy)	0.0.0.0:8042->8042/tcp
ae478b038147	bde2020/hive-metastore-postgresql:2.3.0	"/docker-entrypoint..."	2 days ago	Up 21 minutes	5432/tcp
8902650d80c8	bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	2 days ago	Up 21 minutes (healthy)	0.0.0.0:8188->8188/tcp
3f434cc1df89	mrugankray/resource-manager-python:1.0	"entrypoint.sh /run..."	2 days ago	Up 20 minutes (healthy)	8042/tcp, 0.0.0.0:8088
352f03c553ec	mrugankray/datanode-python:1.0	"entrypoint.sh /run..."	2 days ago	Up 21 minutes (healthy)	0.0.0.0:9864->9864/tcp

Figure 4:Affichage de la liste des conteneurs

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
sentiment_analysis_cluster		Running (12/12)	185.88%		21 minutes ago	
control-center	confluentinc/cp-enterprise-control-c	Running	9.02%	9021-9021	22 minutes ago	
schema-registry	confluentinc/cp-schema-registry:5.4	Running	1.19%	8083-8083	22 minutes ago	
kafka-broker	confluentinc/cp-server:5.4.0	Running	95.5%	29092-29092	22 minutes ago	
hive-server	mrugankray/hive-server-sqoop:1.0	Running	1.56%	10000-10000	22 minutes ago	
zookeeper	confluentinc/cp-zookeeper:5.4.0	Running	0.31%	2181-2181	22 minutes ago	
hive-metastore	bde2020/hive:2.3.2-postgresql-mets	Running	1.02%	9083-9083	22 minutes ago	

Figure 5:Liste des conteneurs

Comme démontré dans la figure 5, tous les services ont été créer avec succès et sont en un état de Running.

Pour accéder à :

- Zeppelin : <http://localhost:8082>
- Namenode : <http://localhost:9000>
- Airflow : <http://localhost:3000>

Remarque :

Assurez-vous que les ports alloués aux conteneurs dans le fichier docker-compose.yml sont vides et accessibles, et ne sont pas utiliser par d'autres applications système.

3. Installation et Configuration de Tableau

Dans le cadre de ce projet, nous utilisons Tableau Desktop. Suivez ces étapes pour installer et configurer Tableau Desktop sur votre machine.

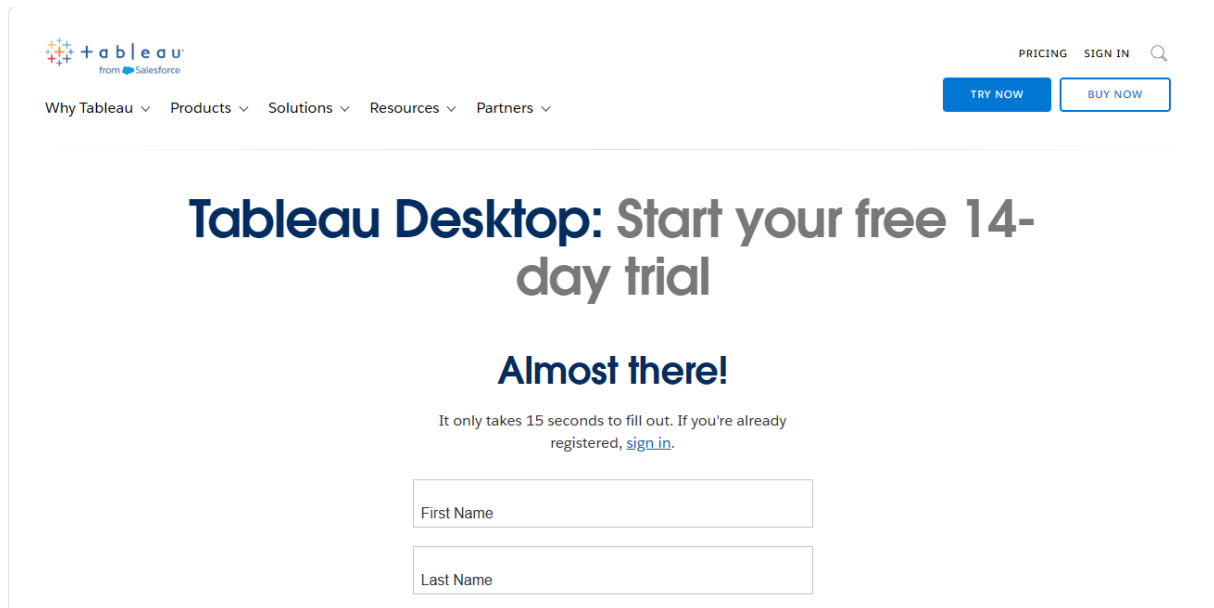
The image shows the Tableau Desktop trial registration page. At the top, there is a navigation bar with the Tableau logo (a colorful grid of dots) and the text '+ a b l e a u from Salesforce'. To the right of the logo are links for 'PRICING', 'SIGN IN', and a search icon. Below the navigation bar is a horizontal menu with links: 'Why Tableau', 'Products', 'Solutions', 'Resources', and 'Partners'. On the right side of the navigation bar, there are two buttons: 'TRY NOW' (solid blue) and 'BUY NOW' (outlined blue). The main content area has a large heading 'Tableau Desktop: Start your free 14-day trial' in a bold, dark blue font. Below this heading is a sub-heading 'Almost there!' in a bold, dark blue font. Underneath the sub-heading is a line of text: 'It only takes 15 seconds to fill out. If you're already registered, [sign in](#).' Below this text are two input fields: 'First Name' and 'Last Name', each with a light gray border and a small 'x' icon in the top right corner. The background is a light gray gradient.

Figure 6: Installation de Tableau Desktop

Téléchargez la version appropriée de Tableau Desktop pour votre système d'exploitation (Windows ou Mac).

Installez Tableau en suivant les instructions fournies lors du processus d'installation

Installation de Hive Tableau Connector

Pour connecter Tableau à Hive, vous pouvez utiliser le pilote Hive Tableau Connector. Suivez ces étapes générales pour effectuer cette connexion.

Téléchargez et installez un pilote compatible avec Hive sur votre machine. Un exemple courant est le pilote Hive Tableau Connector, que vous pouvez trouver sur le site de CDATA

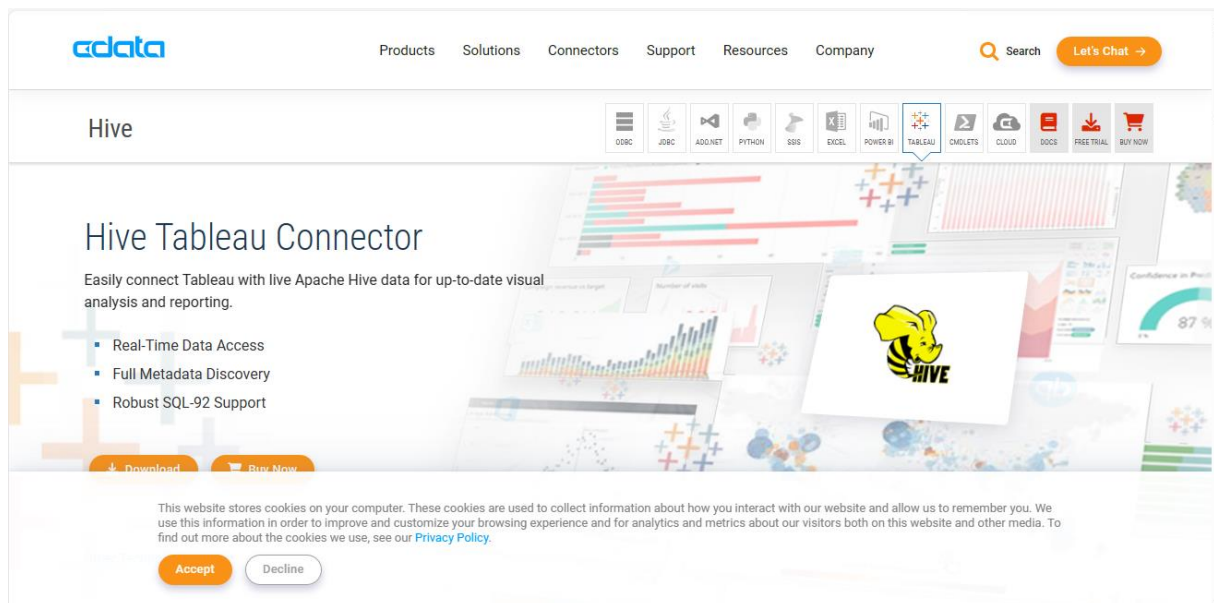


Figure 7: Site de téléchargement de Hive Tableau connector

III. Scraping des données avec NtScraper vers un topic Kafka

Ce script Python effectue le scraping de données depuis Nitter en utilisant la bibliothèque ntscraper, puis envoie ces données à un topic Kafka. Voici un guide rapide pour comprendre le script, avec l'idée qu'il sera intégré dans un notebook Zeppelin :

Importation des bibliothèques :

json: Manipulation de données JSON.

KafkaProducer de la bibliothèque kafka-python : Envoi de messages à Kafka.

Nitter de la bibliothèque ntscraper : Extraction de tweets depuis Nitter.

Définition des configurations Kafka :

KAFKA_BOOTSTRAP_SERVERS: Serveurs Kafka pour la connexion.

KAFKA_TOPIC_NAME: Nom du topic Kafka.

KAFKA_PRODUCER_CONFIG: Configuration du producteur Kafka.

Définition des termes de recherche et continents :

terms: Termes de recherche, par exemple, ["genocide", "gaza", "world"].

continents: Liste de continents, par exemple, ["Africa", "Asia", "Europe", ...].

```

!python
import json
from kafka import KafkaProducer
from ntscraper import Nitter

# Define the Kafka Producer configuration
KAFKA_BOOTSTRAP_SERVERS = ['kafka-broker:29092']
KAFKA_TOPIC_NAME = 'sentiment_analysis'
KAFKA_PRODUCER_CONFIG = {
    'bootstrap_servers': KAFKA_BOOTSTRAP_SERVERS
}

terms = ['genocide', 'gaza', 'world']
continents = ['Africa', 'Asia', 'Europe', 'North America', 'South America']

def get_twitter_data(terms, continents):
    Twitter_data_list = []
    scraper = Nitter()

    for term in terms:
        for country in continents:
            tweets = scraper.get_tweets(term, mode='term', language='en', number=100, near=country)

            # Print information about the tweets variable
            # print(f"Term: {term}, Country: {country}, Tweets: {tweets}")

            for x in tweets['tweets']:
                data = {
                    'text': x['text'],
                    'date': x['date'],
                    'likes': x['stats']['likes'],
                    'is_retweet': x['is-retweet'],
                    'retweets': x['stats']['retweets'],
                    'country': country # Add the country name to the data
                }
                Twitter_data_list.append(data)

    return Twitter_data_list

def main(terms):
    try:
        producer = KafkaProducer(**KAFKA_PRODUCER_CONFIG)

        twitter_data = get_twitter_data(terms, continents)
        for tweet in twitter_data:
            json_data = json.dumps(tweet)
            print(json_data)
            producer.send(KAFKA_TOPIC_NAME, json_data.encode())

    except Exception as e:
        print(f"Error: {e}")

main(terms)

Testing instances: 100%|#####| 29/29 [01:11:00:00, 2.46s/it]

```

Figure 8:Code de Scraping des données avec NtScraper vers un topic Kafka

Note pour Zeppelin :

Lorsque vous exécutez ce script dans un notebook Zeppelin, assurez-vous que les dépendances nécessaires sont installées sur votre environnement Zeppelin.

Assurez-vous que Kafka est en cours d'exécution et accessible depuis Zeppelin.

Le figure 9 represente les tweets scrapé après avoir exécuter le notebook Zeppelin.

```

Testing instances: 100%|#####| 31/31 [01:00:00:00, 1.95s/it]
Testing instances: 100%|#####| 31/31 [01:01:00:00, 1.98s/it]
{"text": "Genocide has a specific meaning and you are misusing the word. I know it is convenient to exaggerate, for brevity and impact, but many people will distrust what you are saying if you misuse such words.", "date": "Dec 16, 2023 \u00b7 10:42 AM UTC", "likes": 0}
{"text": "yet y\u0219all argue everyday on this app that her posting about the palestinian genocide would\u0219t change anything", "date": "Dec 15, 2023 \u00b7 6:12 PM UTC", "likes": 52241}
{"text": "Tragedy Needs No Words: This deaf Gaza is expressing the horrors he's witnessed throughout the ongoing Israeli genocide against the people of Gaza.", "date": "Dec 16, 2023 \u00b7 2:00 AM UTC", "likes": 3527}
{"text": "ITS BEEN 75 YEARS OF GENOCIDE.", "date": "Dec 16, 2023 \u00b7 2:51 AM UTC", "likes": 7171}
{"text": "The very idea that the national broadcaster has adopted a \u0219do not mention the genocide\u0219 approach instead of reporting on genocide as it\u0219s happening should concern everyone who cares about democ racy, truth or just reality.", "date": "Dec 16, 2023 \u00b7 9:15 AM UTC", "likes": 140}
{"text": "Accusations of \"genocide\" is the award Jews get for refusing to die willingly. \u0083e\u00d37\u0200d\u02642\u00e8f", "date": "Dec 15, 2023 \u00b7 1:53 PM UTC", "likes": 79}
{"text": "\u0219Don't photograph us, don't capture images,the world indifferent\u0219d The same sentiments echo from grandmothers in the wars of \u021976 & current genocide on Gaza as the tent is pitched again. The p ain remains unchanged,world across generations,still watches& follows @palestinian_the", "date": "Dec 15, 2023 \u00b7 12:00 PM UTC", "likes": 745}
{"text": "Maybe this can help for Ayden, the genocide sympathizer...", "date": "Dec 16, 2023 \u00b7 10:42 AM UTC", "likes": 0}
{"text": "Genocide is not a trend. Palestine is not a trend. Don\u0219t stop speaking up for the oppressed. Continue demanding for a ceasefire. Keep talking about Palestine.", "date": "Dec 16, 2023 \u00b7 4:56 AM UT C", "likes": 6582}
{"text": "Imagine if genocide made you this happy", "date": "Dec 16, 2023 \u00b7 12:37 AM UTC", "likes": 6251}

```

Figure 9:les tweets scrapé

Les données ou bien les messages peuvent être visualiser au niveau du control center à l'adresse (<http://localhost:9021>)

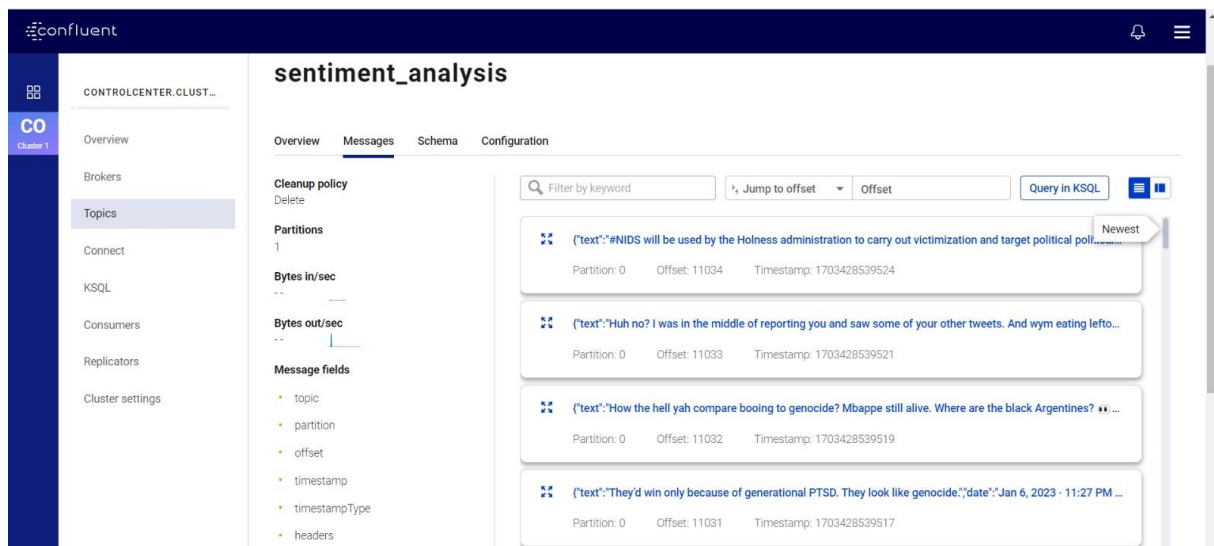


Figure 10:Interface de control center

IV. Traitement et Stockage des Données avec Spark et Hive en format Parquet

Pour effectuer le traitement et le stockage des données avec Spark et Hive en utilisant PySpark dans Zeppelin, vous pouvez suivre ces étapes :

Assurez-vous que votre environnement est correctement configuré pour utiliser PySpark avec Zeppelin et que tout fonctionne correctement.

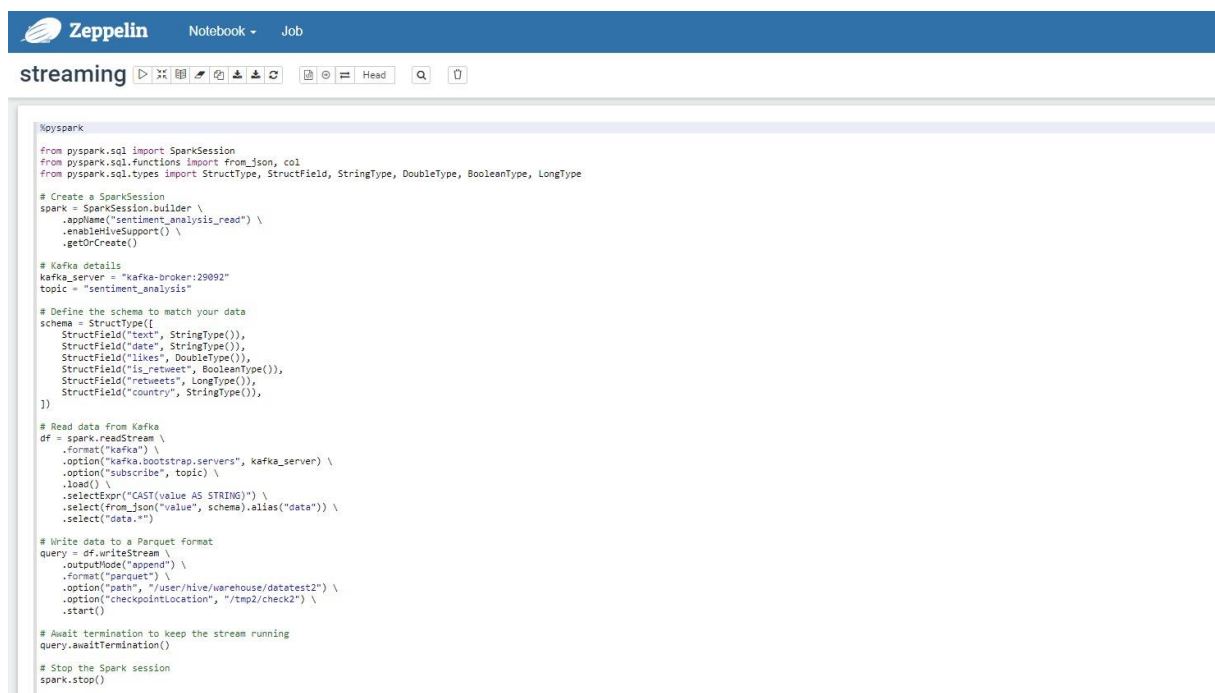
Le script de figure X PySpark orchestre le traitement en continu de données provenant d'un topic Kafka, les transforme conformément à un schéma défini, puis les stocke au format Parquet dans Hive. Il démarre une session Spark, spécifie les détails de Kafka, définit un schéma pour les données, lit les données en continu depuis Kafka, et les écrit en continu dans Hive au format Parquet. L'exécution continue jusqu'à ce qu'elle soit interrompue. Cette approche permet un traitement efficace des flux de données en temps réel, fournissant une structuration optimale et une conservation dans Hive pour une analyse ultérieure. Une fois le traitement terminé, la session Spark est arrêtée.

Avant de lancer l'exécution de votre code (Streaming) créer une table dataset2 dans Hive, qui va contenir les données. Par défaut cette table sera enregistré dans la base de données default: la figure x représente le script de création de la table.

```

hive> CREATE TABLE IF NOT EXISTS datatest2 (
>   text STRING,
>   `date` STRING,
>   likes DOUBLE,
>   is_retweet BOOLEAN,
>   retweets BIGINT,
>   country STRING
> )
> STORED AS PARQUET;
OK
Time taken: 7.621 seconds
hive>
> show tables;
OK
datatest
datatest2
parquet_table_name
Time taken: 0.159 seconds, Fetched: 3 row(s)
hive>

```



The image shows the Zeppelin Notebook interface. At the top, there's a blue header with the Zeppelin logo and the text "Notebook" and "Job". Below the header, the notebook title is "streaming". The main area displays a PySpark script that reads data from a Kafka topic named "sentiment_analysis" and writes it to a Parquet table named "datatest2". The script includes comments for each step: creating a SparkSession, defining Kafka details, defining the schema, reading data from Kafka, writing data to Parquet, and stopping the session.

```

%pyspark

from pyspark.sql import SparkSession
from pyspark.sql.functions import from_json, col
from pyspark.sql.types import StructType, StructField, StringType, DoubleType, BooleanType, LongType

# Create a SparkSession
spark = SparkSession.builder \
    .appName("sentiment_analysis_read") \
    .enableHiveSupport() \
    .getOrCreate()

# Kafka details
kafka_server = "kafka-broker:29092"
topic = "sentiment_analysis"

# Define the schema to match your data
schema = StructType([
    StructField("text", StringType()),
    StructField("date", StringType()),
    StructField("likes", DoubleType()),
    StructField("is_retweet", BooleanType()),
    StructField("retweets", LongType()),
    StructField("country", StringType()),
])

# Read data from Kafka
df = spark.readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", kafka_server) \
    .option("subscribe", topic) \
    .load() \
    .selectExpr("CAST(value AS STRING)") \
    .select(from_json("value", schema).alias("data")) \
    .select("data.*")

# Write data to a Parquet format
query = df.writeStream \
    .outputMode("append") \
    .format("parquet") \
    .option("path", "/user/hive/warehouse/datatest2") \
    .option("checkpointLocation", "/tmp2/check2") \
    .start()

# Await termination to keep the stream running
query.awaitTermination()

# Stop the Spark session
spark.stop()

```

Donc après le lancement de notre script on a dans cette figure x ,notre données sont enregistrée avec succès.

```
hive> select * from dataset2;
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'dataset2'
hive> select * from dataset2;
OK
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
I can't even find words.. STOP THIS GENOCIDE FREE PALESTINE Dec 16, 2023 · 7:53 PM UTC 0.0 NULL 0 Africa
What do you guys gain really from this bullshit.... election is over, you've collected your money. Go find something meaningful to do. The genocide you guys are looking f
or, you will find it o. Ask Rwanda Dec 16, 2023 · 4:56 PM UTC 0.0 NULL 0 Africa
Since the start of Zionist genocide in Gaza, they've been claiming to uncover dead Israeli hostage bodies.....it turns out Zionists are killing their own people in G
aza. The only alive ones they've gotten them through a deal with Hamas. #IsraelIsATerroristState Dec 16, 2023 · 1:11 PM UTC 1.0 NULL 1 Africa
Anti-semitism is a word some terrorists have been using to justify genocide , occupation and outright terror on innocent people, miss us with that bullshit son. De
c 16, 2023 · 12:27 PM UTC 0.0 NULL 0 Africa
You just realize how secondary a lot of things are compared to a genocide in real time Dec 16, 2023 · 8:00 AM UTC 1.0 NULL 0 Africa
Genocide Joe. Enough is enough. Dec 16, 2023 · 1:24 AM UTC 0.0 NULL 0 Africa
Shame to the West for allowing this Genocide to go on unchallenged Dec 15, 2023 · 10:25 PM UTC 0.0 NULL 0 Africa
#Pakistan #Bahrain #Cyprus #Britain #Oman Have #Palestine Genocide blood on their hands. Dec 15, 2023 · 10:20 PM UTC 0.0 NULL 0 Africa
Define "we". I for one is not in support of genocide. #FreePalestineFromIsraelNOW #CeasefireForGazaNOW Dec 15, 2023 · 8:54 PM UTC 0.0 NULL 0 Africa
May Allah accept you among martyrs with your family in his highest Jannah.. END THIS GENOCIDE PERMANENT CEASEFIRE NOW Dec 15, 2023 · 6:40 PM UTC 0.0 NULL 0A
frica
Oh my God 🙏END THIS MADNESS END THIS GENOCIDE Dec 15, 2023 · 6:35 PM UTC 0.0 NULL 0 Africa
STOP GENOCIDE PERMANENT CEASEFIRE NOW FREE PALESTINE Dec 15, 2023 · 3:45 PM UTC 0.0 NULL 0 Africa
It doesn't take rocket science to recognize a genocide.....Israel is committing a genocide with the American and western finances. #IsraelIsATerroristState De
```

Vérification au niveau de Hive

```
A Blue ❤️ Dec 16, 2023 · 10:25 PM UTC 2.0 false 0 South America
if some how what i hear from the people of hawaii happen, ill lose faith in this country and the world forever. Easily letting money and greed make malice, having no reg
ard for the lives of others. easily None of this could be true, but this world doesn't make it far fetched Dec 16, 2023 · 10:24 PM UTC 0.0 false 0 So
uth America
They sold the world for war Dec 16, 2023 · 10:24 PM UTC 0.0 false 0 South America
Me and Mike are working out the future of the world. It's awesome. Dec 16, 2023 · 10:24 PM UTC 0.0 false 0 South America
Justin Fields might've won a Super Bowl by now if he had a good coach 🤔 #Bears | #NFL Dec 16, 2023 · 10:22 PM UTC 1.0 false 0 South America
My 🌟 review of Leave the World Behind on @letterboxd: https://boxd.it/5KXdkJ Dec 16, 2023 · 10:22 PM UTC 0.0 false 0 South America
#ahorasuena Prince & the Revolution, "Around the World in a Day" Mucho mejor de lo que recordaba. Dec 16, 2023 · 10:19 PM UTC 0.0 false 0 South Amer
ica
@McFaul because you know, engaging in the financing of both sides for 2 new wars after a President who lead world peace and no new wars is the right national security pos
ition. And, it's amazing how the failed policies of the Obama/Biden Administration are the goto position as if 8 years of failing hadn't proven those people wrong. Now
we get to watch another 4 years of National Security failure from the wrong people being returned to fail again. The positive though, is Democrats love them some Congres
sional Money Laundering, and they are making use it now. 🍷for Congress is pocketing Billions into the personal accounts as the standard screw the #Taxpayer legislat
ion continues unabated. #TaxAndSpend, #PrintAndSpend, #BorrowAndSpend, and #UseWithoutReplacing @MarshaBlackburn @LaurenBoebert @KatieHobbs @KarlLake @SenatorSinema @kyr
stenisnema @CaptMarkKelly @TheDemocrats @AOC @RepAOC @GOP @POTUS @JoeBiden @MHCOS @VP @WhiteHouse @GOP @RNCResearch @GOPChairwoman @AZGOP @KarlLake @AZSenateGOP @AZHouseG
OP @VivekGrahamswamy @NikkHaley @RonDeSantis @SenatorTimScott @votetimsconfig @RepDavid @RepEliCrane @RepAndyBiggsAZ @JuanCiscomani @RepDLesko @DrPaulGosar @catturd2 @PapiT
rumpo @SecGranholm @southwestpolicy @LeaderMcConnell @SenSchumer @SpeakerJohnson @RepJeffries Dec 16, 2023 · 10:16 PM UTC 0.0 false 0 South America
As an African that has observed all the atrocities the continent has endured since the 90s (while the west simply watched, in Rwanda's case they even debated the definiti
on of "genocide" just so they could further debate if the situation fit) I completely understand/agree her Dec 16, 2023 · 9:46 PM UTC 1.0 NULL 0 Af
rica
Time taken: 2.53 seconds, Fetched: 3455 row(s)
```

Vérification au niveau de Hive :

Pour vérifier que les données ont été correctement stockées dans Hive, accéder à l’interface de Hadoop.On remarque que les données sont un format parquet.

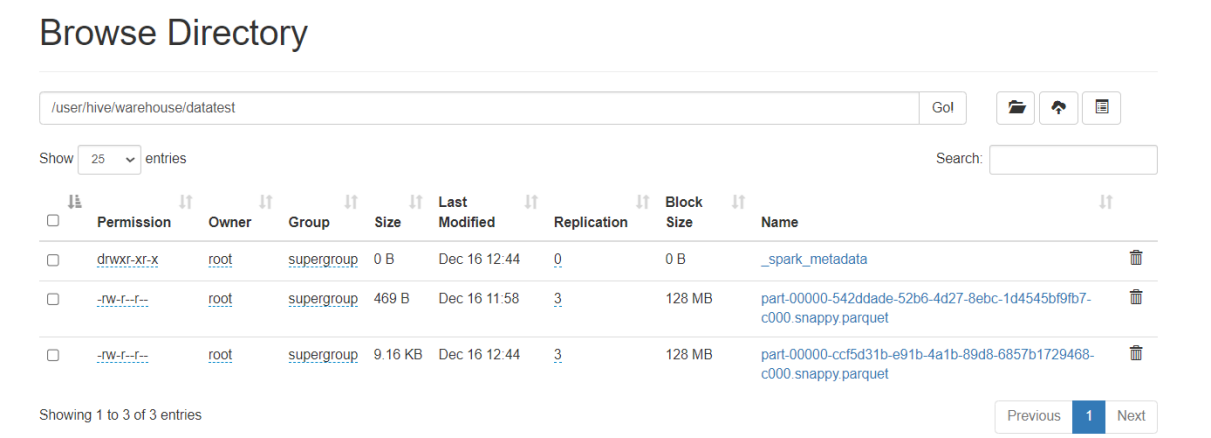


Figure 11:L’ajout des données sur Hive en format parquet

V. Entraînement de modèle avec Spark NLP

Spark NLP est une bibliothèque open-source développée par John Snow Labs, qui fournit des outils avancés pour le traitement du langage naturel (NLP) dans l'écosystème Apache Spark. Il

s'agit d'une extension puissante de la plateforme Spark, exploitant les capacités de traitement distribué de Spark pour réaliser des tâches complexes de NLP à grande échelle. Cette bibliothèque offre une combinaison de performances, d'évolutivité et de facilité d'utilisation pour les applications de traitement de texte dans des environnements distribués.

Dans le notebook suivant, nous allons explorer le potentiel de Spark NLP pour l'analyse de sentiment. Nous allons entraîner un modèle permettant de prédire le sentiment associé à des textes, en utilisant les fonctionnalités avancées de Spark NLP. Cela nous permettra de bénéficier de l'efficacité du traitement distribué de Spark pour entraîner des modèles de NLP performants.

Le code qui suit dans le notebook détaille les étapes de configuration de l'environnement Spark, le chargement des données, l'initialisation de Spark NLP, et enfin, l'utilisation d'un modèle pré-entraîné pour l'analyse de sentiment.

```
!pip install -q pyspark findspark

Preparing metadata (setup.py) ... done
Building wheel for pyspark (setup.py) ... done

[ ] # Not always necessary, but just in case...
import findspark
findspark.init()

CUSTOM_SPARK_SESSION = True

# Common method to create Spark session
from pyspark.sql import SparkSession

if not CUSTOM_SPARK_SESSION:
    spark = SparkSession.builder\
        .master("local[*]")\
        .appName("Colab")\
        .config('spark.ui.port', '4050')\
        .getOrCreate()
    print(f"Spark version: {spark.version}")

[ ] !pip install -q spark-nlp==4.2.4

448.4/448.4 kB 2.9 MB/s eta 0:00:00

[ ] # Because spark-nlp relies on jars, use this function to load them when creating a session.
from pyspark.sql import SparkSession

SPARK_JARS = ["com.johnsnowlabs.nlp:spark-nlp_2.12:4.2.4"]

def get_spark(master="local[*]", name="Colab"):
    builder = SparkSession.builder.appName(name)
    builder.config('spark.ui.port', '4050')
    builder.config('spark.jars.packages', ",".join(SPARK_JARS))
    builder.config("spark.driver.memory", "16G")
    builder.config("spark.serializer", "org.apache.spark.serializer.KryoSerializer")
    builder.config("spark.kryo.serializer.buffer.max", "2000M")
    builder.config("spark.driver.maxResultSize", "0")
    return builder.getOrCreate()

if CUSTOM_SPARK_SESSION:
    spark = get_spark()
    print(f"Spark version: {spark.version}")

Spark version: 3.5.0
```

```
[ ] import sparknlp

from pyspark.ml import Pipeline
from sparknlp.pretrained import PretrainedPipeline
from sparknlp.base import *
from sparknlp.annotator import *

import pandas as pd

# This start() is ignored if a Spark session exists.
# This creates a new spark session if one has not been created previously.
# Additionally, it only loads the jar for spark-nlp, which is a problem if you want to load other jars.
# https://github.com/JohnSnowLabs/spark-nlp/blob/master/src/main/scala/com/johnsnowlabs/nlp/SparkNLP.scala
spark = sparknlp.start()

print("Spark NLP version", sparknlp.version())
print("Apache Spark version:", spark.version)

Spark NLP version 4.2.4
Apache Spark version: 3.5.0

[ ] !wget -O IMDB-Dataset.csv https://github.com/Ankit152/IMDB-sentiment-analysis/blob/master/IMDB-Dataset.csv?raw=true

data = spark.read.csv("IMDB-Dataset.csv", inferSchema=True, header=True, mode='DROPMALFORMED')
data = data.withColumnRenamed('review', 'text').withColumnRenamed('sentiment', 'sentiment_label')
```

text	sentiment_label
One of the other reviewers ...	positive
Basically there's a family ...	negative
I sure would like to see a ...	positive
This show was an amazing, f...	negative
Encouraged by the positive ...	negative
If you like original gut wr...	positive
"Phil the Alien is one of t...	negative
I saw this movie when I was...	negative
The cast played Shakespeare...	negative
This a fantastic movie of t...	positive
Kind of drawn in by the ero...	negative
Some films just simply shou...	positive
This movie made it into one...	negative
I remember this film,it was...	positive
An awful film! It must have...	negative
After the success of Die Ha...	positive
What an absolutely stunning...	positive
This was the worst movie I ...	negative
The Karen Carpenter Story s...	positive
This film tried to be too m...	negative

only showing top 20 rows

Ce code en Spark NLP définit et exécute un pipeline de traitement du langage naturel (NLP) en utilisant Spark. Voici une explication détaillée de chaque composant du pipeline :

1.DocumentAssembler :

- Cette étape prépare le texte pour le traitement en le convertissant en un objet Document Spark NLP.
- **InputCol :** "text" (colonne d'entrée du texte).
- **OutputCol :** "document" (colonne de sortie contenant le document préparé).

2. Tokenizer :

- Cette étape divise le document en tokens (mots).
- **InputCols :** ["document"] (colonne d'entrée contenant le document préparé).
- **OutputCol :** "token" (colonne de sortie contenant les tokens).

3. Normalizer :

- Cette étape normalise les tokens, les préparant pour l'analyse ultérieure.
- **InputCols** : ["token"] (colonne d'entrée contenant les tokens).
- **OutputCol** : "normal" (colonne de sortie contenant les tokens normalisés).

4. ViveknSentimentModel :

- C'est un modèle pré-entraîné pour l'analyse de sentiment basé sur l'algorithme Vivekn.
- **InputCols** : ["document", "normal"] (colonnes d'entrée contenant le document préparé et les tokens normalisés).
- **OutputCol** : "result_sentiment" (colonne de sortie contenant les résultats de l'analyse de sentiment).

5. Finisher :

- Cette étape finalise les résultats, produisant une colonne de sortie contenant les sentiments sous une forme plus conviviale.
- **InputCols** : ["result_sentiment"] (colonne d'entrée contenant les résultats de l'analyse de sentiment).
- **OutputCols** : "final_sentiment" (colonne de sortie contenant les sentiments finalisés).

Pipeline :

- La pipeline est assemblée en définissant l'ordre des étapes.
- **setStages** : Les différentes étapes de la pipeline sont spécifiées dans l'ordre.

Fit and Transform :

- La pipeline est ajustée (fit) aux données, puis les données sont transformées (transform) en utilisant le pipeline.

En résumé, ce pipeline effectue une analyse de sentiment sur des données textuelles en utilisant le modèle pré-entraîné Vivekn dans le cadre de Spark NLP. Le résultat final est un DataFrame contenant une colonne de sentiments analysés.


```
[ ] # common text to check
comment = "The movie I watched today was not a good one"
```

```
# https://nlp.johnsnowlabs.com/2021/11/22/sentiment_vivekn_en.html

document = DocumentAssembler() \
.setInputCol("text") \
.setOutputCol("document")

token = Tokenizer() \
.setInputCols(["document"]) \
.setOutputCol("token")

normalizer = Normalizer() \
.setInputCols(["token"]) \
.setOutputCol("normal")

vivekn = ViveknSentimentModel.pretrained() \
.setInputCols(["document", "normal"]) \
.setOutputCol("result_sentiment")

finisher = Finisher() \
.setInputCols(["result_sentiment"]) \
.setOutputCols(["final_sentiment"])

pipeline = Pipeline().setStages([document, token, normalizer, vivekn, finisher])

# Fit to data
pipelineModel = pipeline.fit(data)
result = pipelineModel.transform(data)

result.show(truncate=75)
```

```
Approximate size to download 873.6 KB
[OK!]
```

text	sentiment_label	final_sentiment
[One of the other reviewers has mentioned that after watching just 1 Oz e...	positive	[positive]
Basically there's a family where a little boy (Jake) thinks there's a zo...	negative	[positive]
I sure would like to see a resurrection of a up dated Seahunt series wit...	positive	[positive]
This show was an amazing, fresh & innovative idea in the 70's when it fi...	negative	[positive]
I encouraged by the positive comments about this film on here I was lookin...	negative	[negative]
If you like original gut wrenching laughter you will like this movie. If...	positive	[positive]
I Phil the Alien is one of those quirky films where the humour is based a...	negative	[negative]
I saw this movie when I was about 12 when it came out. I recall the scar...	negative	[positive]
The cast played Shakespeare. Shakespeare lost. I a...	negative	[negative]
This a fantastic movie of three prisoners who become famous. One of the ...	positive	[negative]
I kind of drawn in by the erotic scenes, only to realize this was one of t...	negative	[negative]
Some films just simply should not be remade. This is one of them. In and...	positive	[positive]
This movie made it into one of my top 10 most awful movies. Horrible. <b...	negative	[negative]
I remember this film,it was the first film i had watched at the cinema t...	positive	[positive]
An awful film! It must have been up against some real stinkers to be nom...	negative	[positive]
After the success of Die Hard and it's sequels it's no surprise really t...	positive	[positive]
What an absolutely stunning movie, if you have 2.5 hrs to kill, watch it...	positive	[positive]
This was the worst movie I saw at worldfest and it also received the lea...	negative	[negative]
The Karen Carpenter Story shows a little more about singer Karen Carpent...	positive	[positive]
This film tried to be too many things all at once: stinging political sa...	negative	[negative]

```
only showing top 20 rows
```

```
[ ] # apply trained model to text
example = spark.createDataFrame([comment]).toDF("text")
pipelineModel.transform(example).show(truncate=False)
```

text	final_sentiment
The movie I watched today was not a good one	[negative]

VI. Visualisation des Résultats avec Tableau

La visualisation des résultats avec Tableau nécessite que les données collectées soient stockées dans un format compatible avec Tableau, tel que dans une base de données Hive. Assurez-vous d'avoir suivi les étapes précédentes concernant l'installation de de Tableau Desktop et le pilote Hive Tableau Connector.

Connexion à Hive depuis Tableau :

Après l'installation du pilote, configurez-le en fournissant les détails de connexion à votre cluster Hive, tels que l'adresse du serveur Hive, le port, le nom d'utilisateur, le mot de passe, etc.

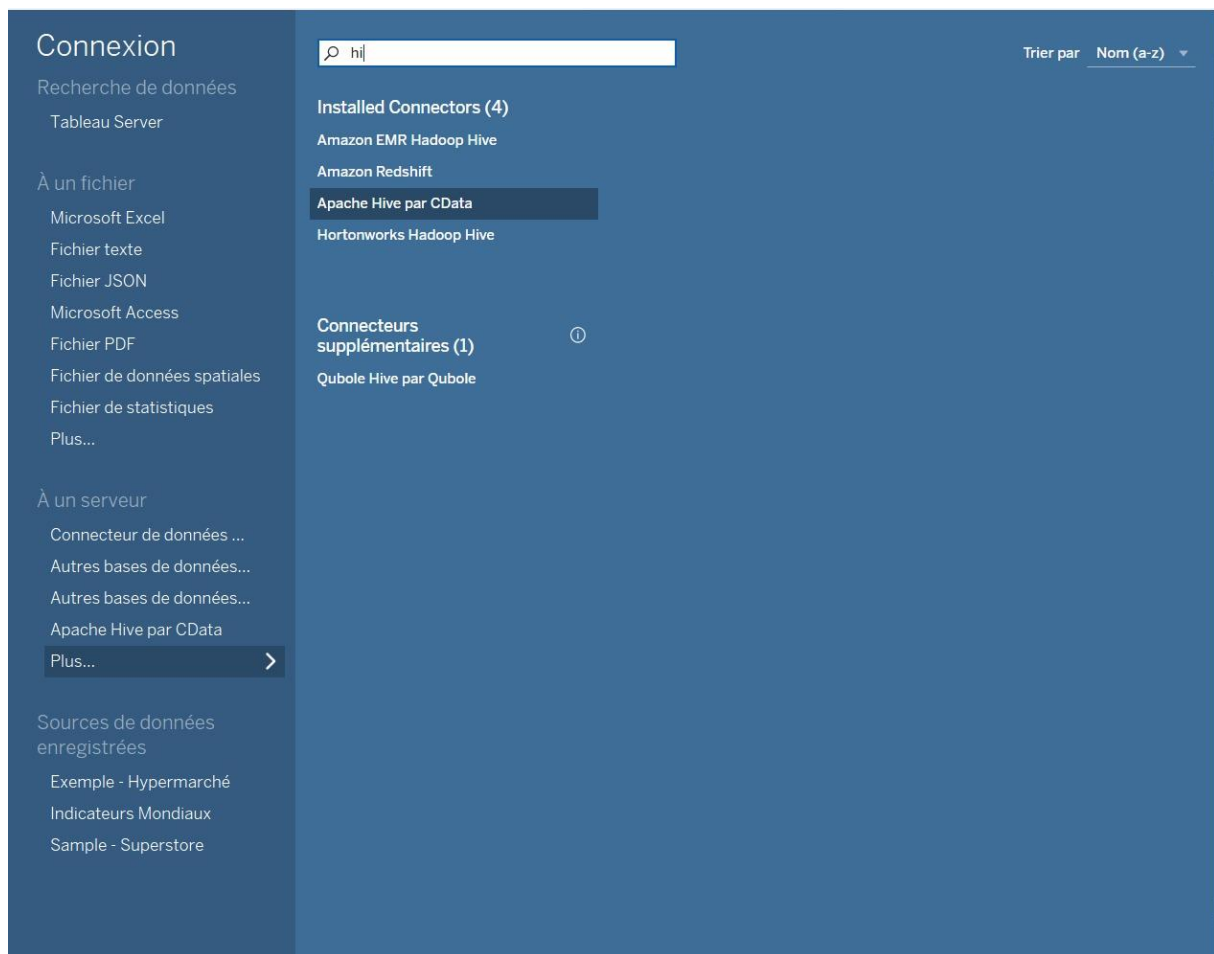


Figure 12: Connexion aux cluster Hive

Dans Tableau Desktop, sous l'onglet "Connexion a un serveur", choisissez "Apache Hive par CData" comme type de connexion.

Sélectionnez le pilote pour Hive que vous avez installé.

Entrez les informations de connexion nécessaires, telles que le nom du serveur Hive, le port, le nom d'utilisateur et le mot de passe.

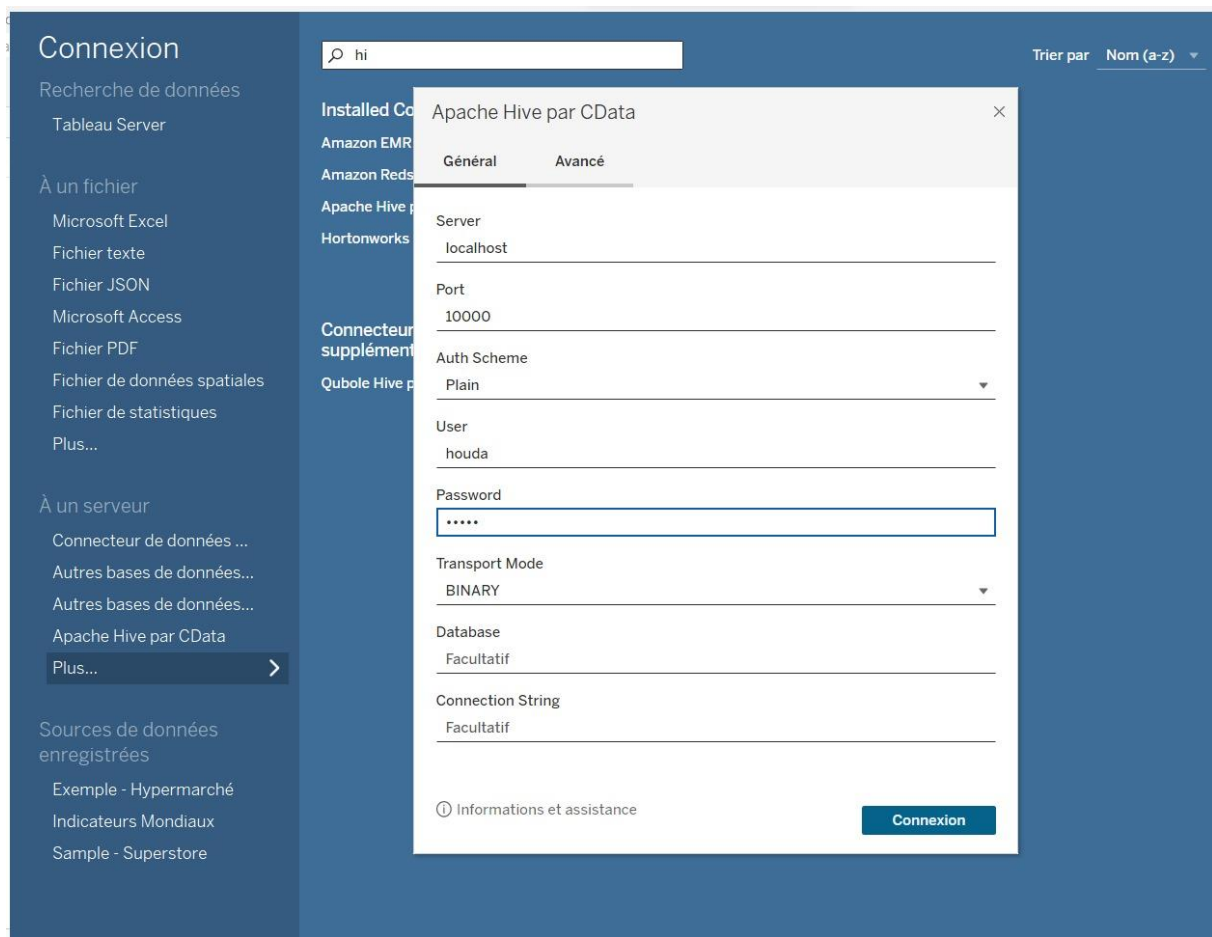


Figure 13: Etablissement de la connexion entre Hive et Tableau

En fonction du pilote que vous utilisez, il peut y avoir des options supplémentaires à configurer. Consultez la documentation du pilote pour plus de détails.

Après avoir configuré les paramètres de connexion, cliquez sur le bouton "Connecter" pour établir la connexion entre Tableau et Hive.

Note : Par défaut, aucun utilisateur n'est configuré sur Hive. Vous devez définir ici un utilisateur et un mot de passe que vous utiliserez à chaque fois que vous établirez la connexion.

Sélectionner la Table ou la Vue Hive :

Une fois connecté, vous pourrez voir les bases de données Hive disponibles. Sélectionnez la base de données, puis choisissez la table ou la vue Hive que vous souhaitez analyser dans Tableau.

Voici comment vous pouvez visualiser les données dans Tableau après avoir suivi les étapes du projet.

Importation des Données

- Connecter vous à nouveau à Hive si nécessaire
- Une fois connecté à Hive, vous verrez les bases de données disponibles.

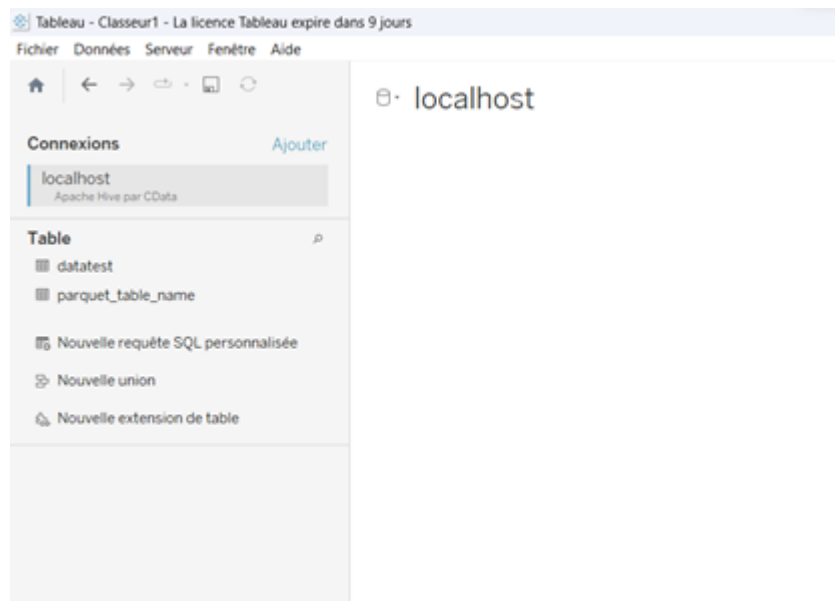



Figure 14:List des Tables

- Sélectionnez la base de données où vous avez stocké les données scrapées.
- Choisissez la table ou la vue qui contient les données scrapées.

dataatest2 (localhost) Connexion: ☒ En direct ☐ Extraire Filtres: 0 | Ajouter

dataatest2



Vous avez besoin de données supplémentaires ?
Faites glisser les tables ici pour les relier. [En savoir plus](#)

dataatest2 6 champs 0 lignes 1000 lignes

Nom	Type	Nom du champ	Table physique	Nom du champ ...
dataatest2	Text		dataatest2	text
	Date		dataatest2	date
	Likes		dataatest2	likes
	Is Retweet		dataatest2	is_retweet
	Retweets		dataatest2	retweets
	Country		dataatest2	country

Text	Date	Likes	Is Retweet	Retweets	Country
We will remember who stood ...	Dec 13, 2023 · 8:09 AM UTC	0.00	Faux	0	Africa
May Allah accept you among ...	Dec 13, 2023 · 7:41 AM UTC	1.00	Faux	0	Africa
Did you know about the little...	Dec 13, 2023 · 6:57 AM UTC	22.00	Faux	5	Africa
You don't care about any of t...	Dec 13, 2023 · 6:19 AM UTC	1.00	Faux	1	Africa
Polish Mp puts out the festiv...	Dec 13, 2023 · 4:47 AM UTC	3.00	Faux	0	Africa
Genocide Joe at it again 🍷	Dec 13, 2023 · 4:05 AM UTC	0.00	Faux	0	Africa
The UN Security Council arti...	Dec 13, 2023 · 3:56 AM UTC	12.00	Faux	4	Africa
When this fool Uncle Tom of ...	Dec 13, 2023 · 12:24 AM UTC	1.00	Faux	0	Africa
Whoever Support Genocide ...	Dec 12, 2023 · 10:34 PM UTC	2.00	Faux	0	Africa
striking through the Auschwi...	Dec 12, 2023 · 9:29 PM UTC	3.00	Faux	1	Africa

Figure 15:Importation des données de Hive sur tableau

Création des Visualisations

Allez dans l'onglet "Feuille" pour commencer à créer vos visualisations.

Sur la gauche, vous verrez une liste de dimensions et de mesures (valeurs numériques). Faites glisser les dimensions et les mesures souhaitées vers les étagères "Colonnes" et "Lignes".

En fonction des données que vous avez, choisissez le type de visualisation adapté. Par exemple, utilisez un diagramme linéaire pour suivre une tendance temporelle, ou une carte pour afficher des données géographiques.

Ajoutez des filtres pour permettre aux utilisateurs de sélectionner des plages de dates, des villes ou d'autres critères spécifiques.

La figure 16 représente graphiquement le nombre de likes et de retweets reçus par pays pour un tweet, utilisant une visualisation en barres pour rendre les données facilement interprétables. Sur l'axe horizontal des X, chaque barre correspond à un pays particulier, disposée selon l'ordre décroissant ou croissant du nombre total de likes et retweets combinés. L'axe vertical des Y quantifie ces interactions, divisant chaque barre en deux sections distinctes : une pour les likes, indiquée en bleu, et une pour les retweets, indiquée en orange. Cette présentation visuelle permet une comparaison rapide de l'engagement dans différents pays, offrant une perspective claire sur les performances du tweet et mettant en évidence les variations significatives d'interaction sur les réseaux sociaux.

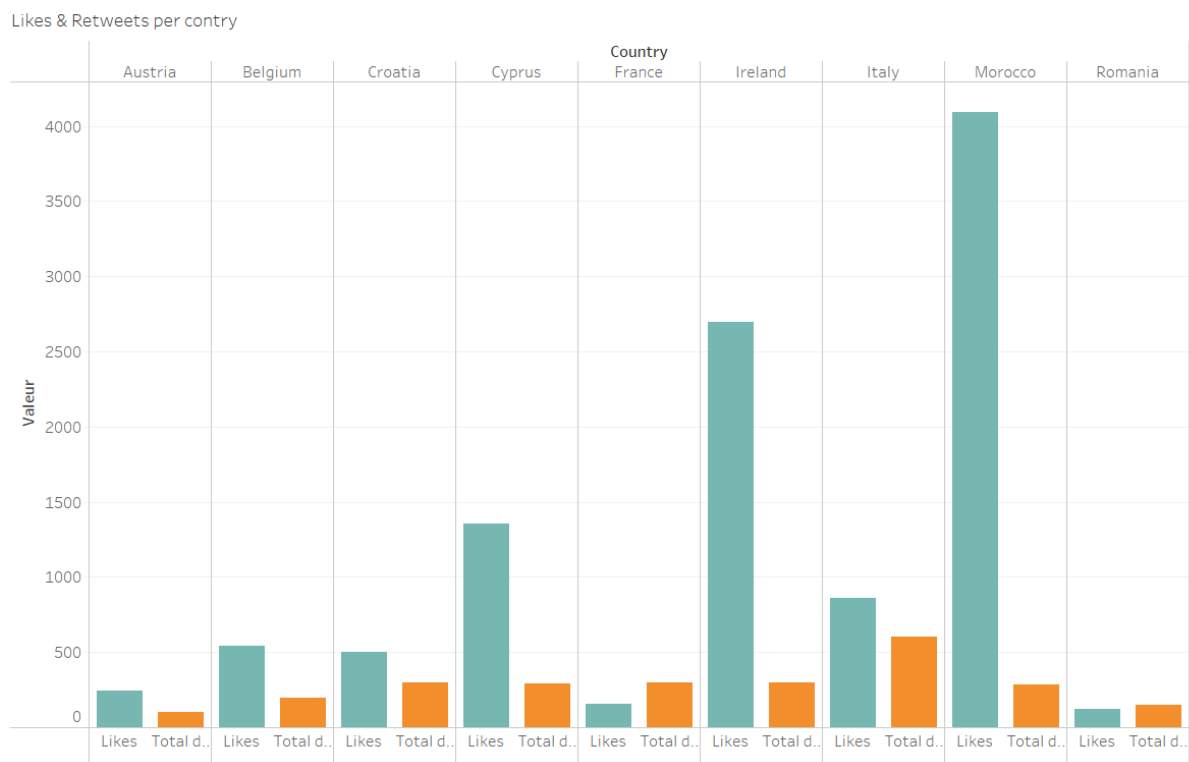


Figure 16:Le nombre de likes et de retweets reçus par pays pour un tweet

Ce graphique représente un graphique à barres inversé utilisé pour illustrer le nombre de tweets dans chaque pays. Les noms des pays sont disposés le long de l'axe des Y (vertical), tandis que le nombre de tweets est représenté sur l'axe des X (horizontal). Chaque barre horizontale correspond à un pays spécifique, et la longueur de chaque barre indique le nombre de tweets émis depuis ce pays. Les barres sont alignées côte à côte le long de l'axe des X. Cette configuration permet une comparaison visuelle rapide du nombre de tweets entre différents

pays, mettant en évidence les variations d'activité sur les réseaux sociaux dans une perspective horizontale.

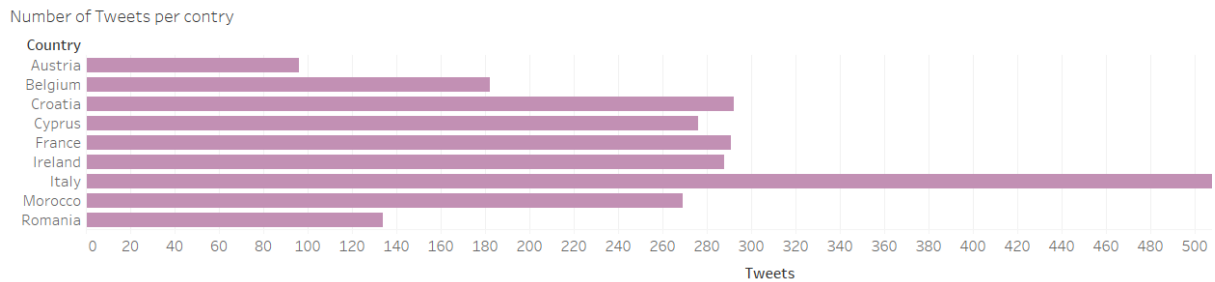


Figure 17:Le nombre de tweets dans chaque pays

La figure 18 illustre la distribution géographique des tweets dans le monde à l'aide d'une carte mondiale. Chaque pays est distingué par des nuances de couleur ou des intensités de teinte, offrant une visualisation claire du volume relatif de tweets émis depuis chaque région.

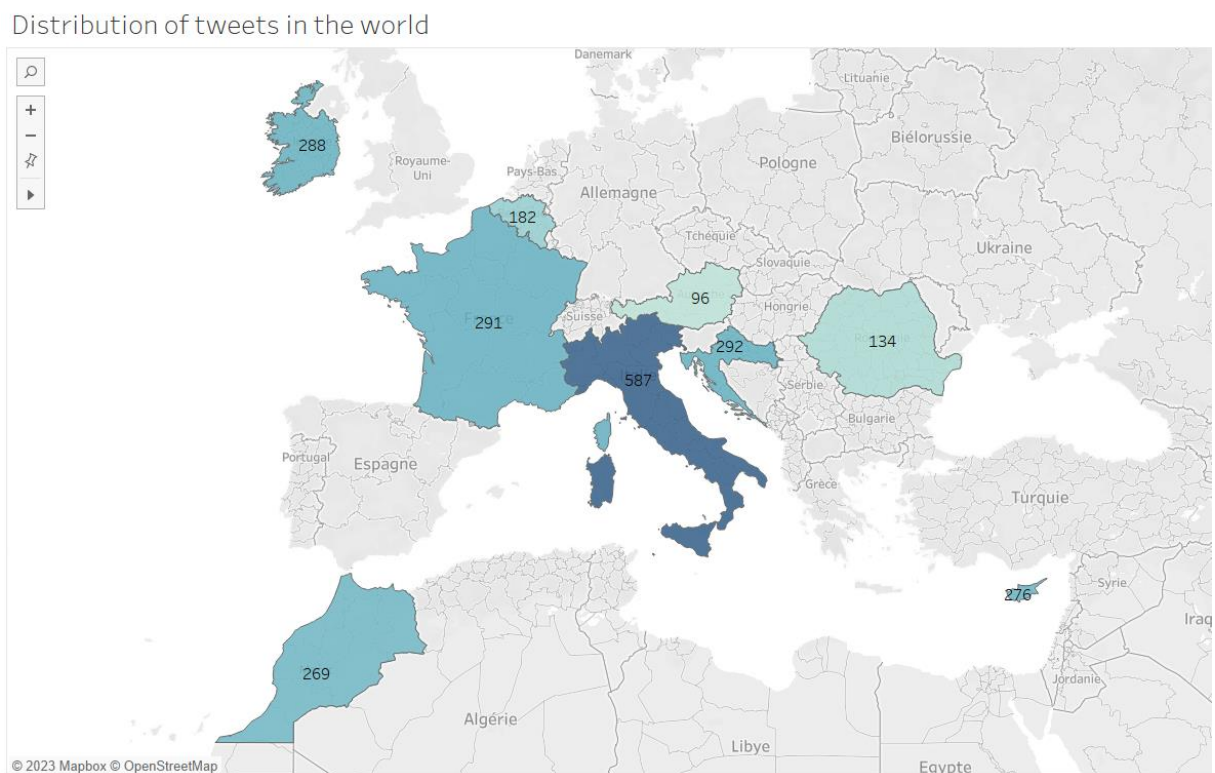


Figure 18:Distribution des tweets dans le monde

La figure 19 offre une vue d'ensemble d'un tableau de bord regroupant une variété de visualisations que nous avons élaborées. Ce tableau de bord englobe différentes représentations graphiques destinées à fournir une analyse approfondie des données.

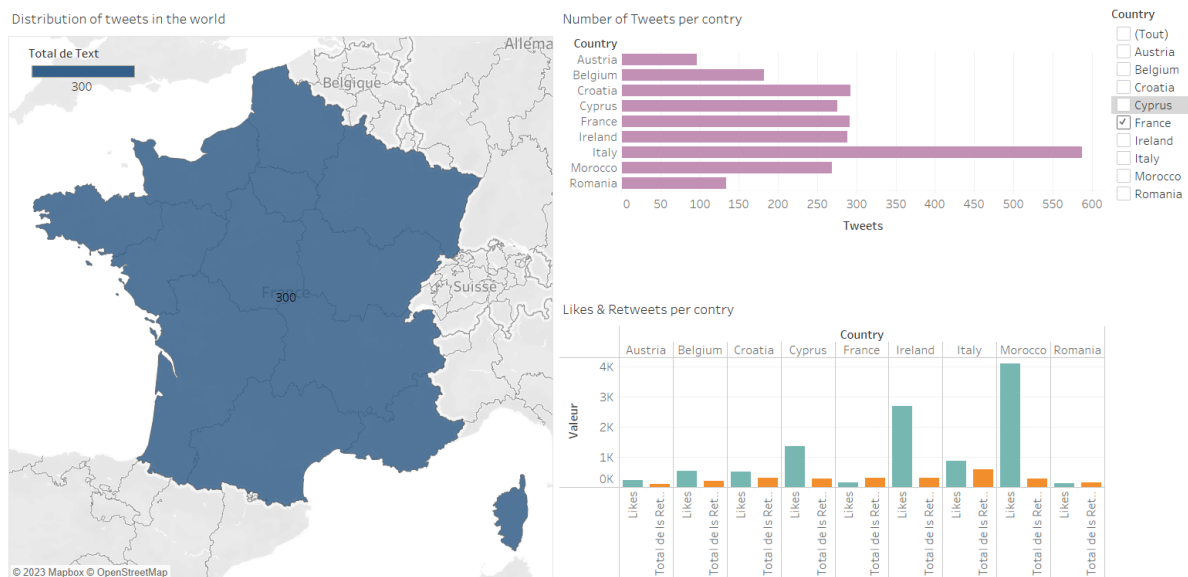


Figure 19:Tableau de bord

VII. Orchestration avec Apache Airflow

Airflow est une plateforme qui vous permet de créer et d'exécuter des flux de travail. Un flux de travail est représenté sous la forme d'un DAG (un graphe acyclique dirigé) et contient des éléments de travail individuels appelés Tâches, organisés avec des dépendances et des flux de données pris en compte.

Un DAG spécifie les dépendances entre les tâches et l'ordre dans lequel les exécuter et exécuter les nouvelles tentatives.

Les tâches elles-mêmes décrivent ce qu'il faut faire, qu'il s'agisse de récupérer des données, d'exécuter une analyse, de déclencher d'autres systèmes, ou plus encore.

Interface utilisateur

Airflow est livré avec une interface utilisateur qui vous permet de voir ce que font les DAG et leurs tâches, de déclencher des exécutions de DAG, d'afficher les journaux et d'effectuer un débogage et une résolution limités des problèmes avec vos DAG.

Pour se connecter à l'interface utilisez les identifiants suivants :

-Nom d'utilisateur : admin

-Mot de passe : admin

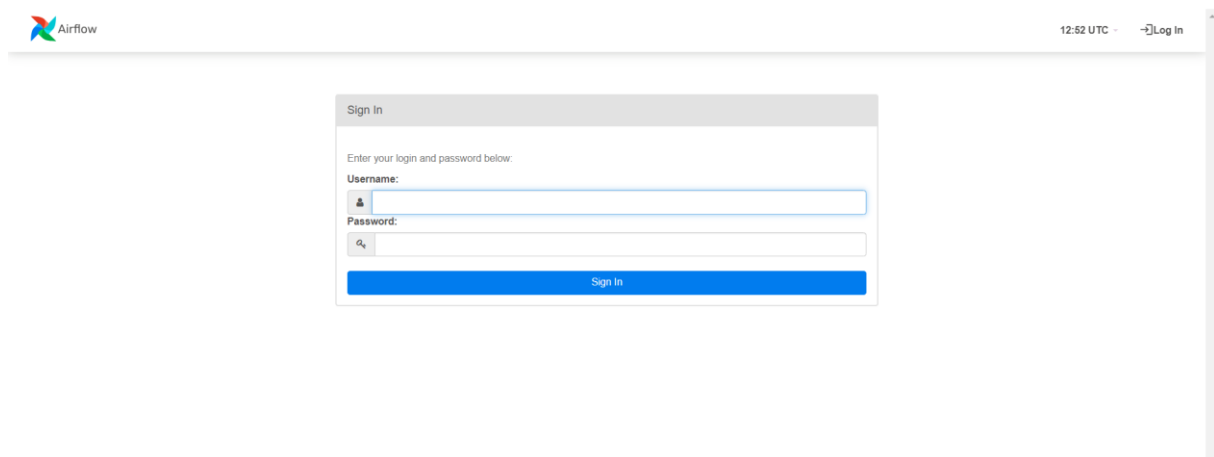


Figure 20: Connexion à l'interface utilisateur

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
Weather_Project_DAG	airflow	0	@daily		2023-12-15, 00:00:00			
example_bash_operator	airflow	0	@daily		2023-12-15, 00:00:00			
example_branch_datetime_operator	airflow	0	@daily		2023-12-15, 00:00:00			
example_branch_datetime_operator_2	airflow	0	@daily		2023-12-15, 00:00:00			
example_branch_dop_operator_v3	airflow	0	@daily		2023-12-16, 12:51:00			
example_branch_labels	airflow	0	@daily		2023-12-15, 00:00:00			
example_branch_operator	airflow	0	@daily		2023-12-15, 00:00:00			
example_branch_python_operator_decorator	airflow	0	@daily		2023-12-15, 00:00:00			
example_complex	airflow	0	None					
example_dag_decorator	airflow	0	None					

Figure 21: Interface utilisateur de Airflow

Création et exécution du dag

- Utilisez votre éditeur de code préféré (comme VSCode, PyCharm, etc.).
- Créez un nouveau dossier appelé "dags" à l'emplacement où vous souhaitez stocker vos DAGs.
- À l'intérieur du dossier "dags", créez un fichier Python pour votre DAG, par exemple, "dag.py".
- Dans ce fichier, écrivez le code décrivant les différentes étapes du processus. Vous trouverez le code dans le fichier dag.py dans le dossier du projet, copier et coller le code dans votre fichier et faites les installations nécessaires.
- Faites toutes les installations de packages nécessaires avec pip ou si vous utilisez PyCharm à partir de l'interpréteur.
- Dans le code du dag modifier les adresses des notebooks selon les id

Ce DAG (Directed Acyclic Graph) créé dans Apache Airflow orchestre le traitement périodique de données Twitter en utilisant deux notebooks Zeppelin.

Les fonctions `execute_zeppelin_notebook_1` et `execute_zeppelin_notebook_2` sont définies pour exécuter respectivement les notebooks Zeppelin associés via des requêtes

POST à leurs URL d'API. Les opérateurs Python `kafka_stream` et `spark_stream` sont ajoutés au DAG pour appeler ces fonctions. Le DAG, nommé 'Twitter_Project_DAG1', est planifié pour s'exécuter toutes les 5 secondes, garantissant un traitement fréquent des données Twitter. De plus, il est configuré pour n'autoriser qu'une seule exécution active à la fois, assurant ainsi l'intégrité du flux de traitement. La tâche `kafka_stream` est définie pour être exécutée en premier, suivie de la tâche `spark_stream`, assurant ainsi un ordre d'exécution cohérent. Ce DAG offre une structure robuste pour automatiser le processus de récupération et de traitement des données Twitter à l'aide de notebooks Zeppelin dans un environnement Airflow.

```
dag.py x
A-big-data-architecture-for-Social-network-based-event-tracking-and-monitoring > dags > dag.py
1 from datetime import timedelta
2 from airflow import DAG
3 from airflow.utils.dates import days_ago
4 from airflow.operators.python import PythonOperator
5 import requests
6
7
8 def execute_zeppelin_notebook_1(**kwargs):
9     # Zeppelin API endpoint to execute a notebook
10    zeppelin_api_url = 'http://localhost:8082/api/notebook/job/2JHG36R93'
11
12    # Make a POST request to execute the Zeppelin notebook
13    response = requests.post(zeppelin_api_url)
14
15    # Check if the request was successful (HTTP status code 200)
16    if response.status_code == 200:
17        print("Zeppelin notebook executed successfully 1.")
18    else:
19        print(f"Failed to execute Zeppelin notebook 1. Status code: {response.status_code}")
20        print(response.text)
21
22
23 def execute_zeppelin_notebook_2(**kwargs):
24     # Zeppelin API endpoint to execute a notebook
25    zeppelin_api_url = 'http://localhost:8082/api/notebook/job/2JHBN3XEZ'
26
27    # Make a POST request to execute the Zeppelin notebook
28    response = requests.post(zeppelin_api_url)
29
30    # Check if the request was successful (HTTP status code 200)
31    if response.status_code == 200:
32        print("Zeppelin notebook executed successfully 2.")
33
```

```
dag.py X
A-big-data-architecture-for-Social-network--based-event-tracking-and-monitoring > dags > dag.py
32     print("Zeppelin notebook executed successfully 2.")
33     else:
34         print(f"Failed to execute Zeppelin notebook 2. Status code: {response.status_code}")
35         print(response.text)
36
37
38
39 # Default DAG arguments
40 default_args = {
41     'owner': 'airflow',
42     'depends_on_past': False,
43     'start_date': days_ago(1),
44     'retries': 1,
45     'retry_delay': timedelta(minutes=5),
46 }
47
48 # Create the main DAG
49 dag = DAG(
50     'Twitter_Project_DAG1',
51     default_args=default_args,
52     description='An Airflow DAG to fetch Twitter data and update the Hive Data',
53     schedule_interval='*/5 * * * *', # Run every 3 hours
54     max_active_runs=1, # Ensure only one run at a time
55     catchup=False, # Do not run backfill for the intervals between start_date and the current date
56 )
57
58 with dag:
59     # Use the PythonOperator to execute the Zeppelin notebook
60     kafka_stream = PythonOperator(
61         task_id='kafka_stream',
62         python_callable=execute_zeppelin_notebook_1,
63         provide_context=True, # Pass the context to the Python function
64
65
66
67     spark_stream = PythonOperator(
68         task_id='spark_stream',
69         python_callable=execute_zeppelin_notebook_2,
70         provide_context=True, # Pass the context to the Python function
71
72
73
74
75 # Set task dependencies
76 kafka_stream >> spark_stream
```

Figure 22:Code du dag du projet

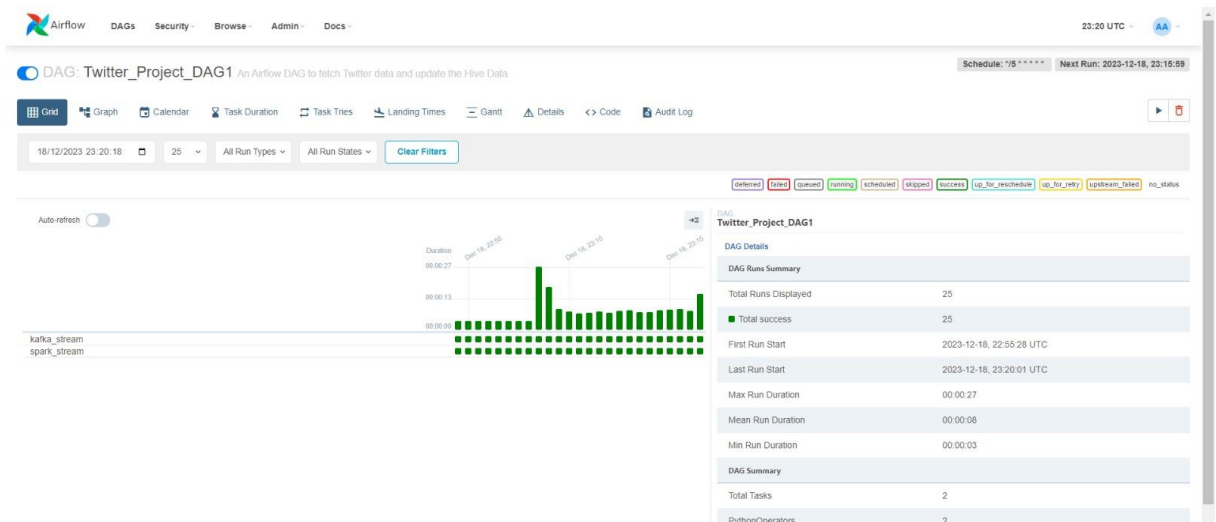


Figure 23: Visualisation de l'exécution du dag créer

Si vous avez planifié le processus de collecte, de traitement et de stockage des données à intervalles réguliers avec Apache Airflow, vous pouvez configurer Tableau pour actualiser automatiquement les données à partir de Hive.