Contexte

Notre entreprise a récemment lancé une application de monitoring météo en temps réel pour permettre aux utilisateurs de suivre l'évolution de la météo à travers différentes régions et villes. Nous utiliserons Kafka pour collecter les données et Spark Streaming pour leur traitement.

Objectifs

- 1. Collecter des données en temps réel à partir de l'API OpenWeatherMap dans un topic Kafka nommé topic-weather.
- 2. Traiter les données en temps réel avec Spark Streaming pour produire de nouvelles variables et stocker les résultats dans un nouveau topic Kafka nommé topic-weather-final.
- 3. Fournir un pipeline fonctionnel et prêt à l'emploi.

Installation et Configuration

Création du dépôt

- 1. Créer un dépôt GitHub nommé projet-esme :
- 2. Ouvrir un codespace sur le dépôt créé.

Installation de Java

1. Mettre à jour les paquets :

sudo	ap	ot-c	et	up	date	Э

_	_	_	_			
2	Inctal	lor.	Java .	אחו	11	
∠.	IIIStai	ıeı	Java	JUN		

sudo apt-get install openjdk-11-jdk-headless

3. Vérifiez l'installation :

java --version

Configuration de Java

Ajouter Java à votre variable d'environnement :

```
echo 'export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64' >> ~/.bashrc echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.bashrc source ~/.bashrc
```

Installation de Kafka

Téléchargez et extrayez Kafka:

wget https://archive.apache.org/dist/kafka/2.6.0/kafka_2.12-2.6.0.tgz tar -xzf kafka_2.12-2.6.0.tgz

Lancement de Zookeeper et Kafka

1. Lancer Zookeeper:

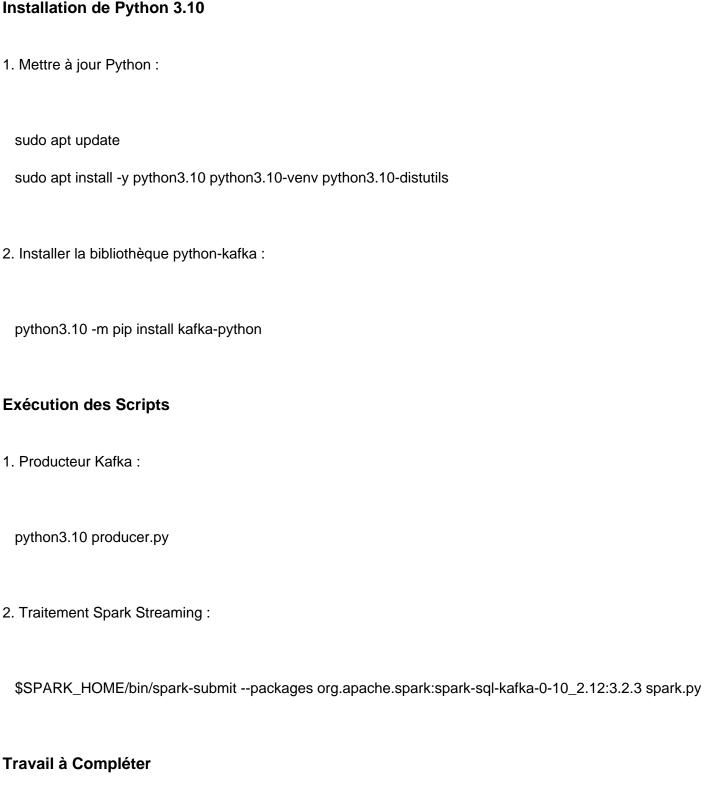
./kafka_2.12-2.6.0/bin/zookeeper-server-start.sh ./kafka_2.12-2.6.0/config/zookeeper.properties
2. Lancer Kafka :
./kafka_2.12-2.6.0/bin/kafka-server-start.sh ./kafka_2.12-2.6.0/config/server.properties
Création de Topics Kafka
1. Créer un topic nommé topic-weather :
./kafka_2.12-2.6.0/bin/kafka-topics.shcreatebootstrap-server localhost:9092replication-factor ofpartitions 1topic topic-weather
2. Créer un topic nommé topic-weather-final :
./kafka_2.12-2.6.0/bin/kafka-topics.shcreatebootstrap-server localhost:9092replication-factor ofpartitions 1topic topic-weather-final
Installation de Spark
Téléchargez et extrayez Spark :
wget https://archive.apache.org/dist/spark/spark-3.2.3/spark-3.2.3-bin-hadoop2.7.tgz tar -xvf spark-3.2.3-bin-hadoop2.7.tgz

export SPARK_HOME=/workspaces/<votre-repertoire>/spark-3.2.3-bin-hadoop2.7

Configurez Spark:

export PATH=\$SPARK_HOME/bin:\$PATH

Installation de P	vthon 3.10
-------------------	------------



1. Compléter le producteur Kafka (producer.py) pour récupérer les données météo depuis l'API

OpenWeatherMap et les envoyer vers topic-weather.

2. Compléter le traitement Spark (spark.py) pour générer de nouvelles variables et envoyer les résultats dans topic-weather-final.
Évaluation
1. Code fonctionnel : Le pipeline doit fonctionner de bout en bout.
2. Documentation : le répo et le code doivent être bien documentés.
3. Présentation orale : Présenter le projet. Des questions vous seront posées.
Bonne chance !