PROJET BIG DATA

Présenté à la Faculté des Sciences de Sfax

Elaboré par

Yessine Karray

Architecture Big Data Distribuée pour l'Analyse des Risques Médicamenteux

Soutenu le 20/12/2024

Mr Mohamed Ali HadjTaib	Prof du cours
Mr Montassar Akremi	Prof du TP

Table des Matières

1. Introduction
2. Vue d'Ensemble du Projet

❖ **1ère approche (Kafka+Spark+Hive+Hadoop+superset)**

3. Architecture Technique
4. Configuration Détaillée
5. Des captures D'ecrans

❖ **2 éme approche (Spark +Doris+Superset)**

6. Architecture Technique
7. Configuration Détaillée
8. Des captures D'ecrans

1. Introduction

1. Description

L'objectif de ce projet est d'exploiter les informations fournies par l'API OpenFDA afin d'examiner les rapports de sécurité concernant les médicaments pharmaceutiques. En analysant les effets secondaires, les rappels de médicaments et les données de sécurité, l'étude repérera les tendances des effets secondaires rapportés ainsi que les facteurs de risque potentiels liés à différents médicaments.

Les résultats contribueront à une meilleure compréhension des risques potentiels liés à certains médicaments pour les professionnels de santé, les autorités de réglementation et les patients.

2. Objectifs

- **Surveiller les tendances de la sécurité des médicaments :** Identifier les tendances des événements indésirables signalés pour des médicaments spécifiques, tels que les effets secondaires couramment signalés.

- **Identifier les médicaments à haut risque :** Mettez en évidence les médicaments avec un grand nombre de rapports d'effets indésirables, en particulier ceux liés à des issues de santé graves.

- **Soutenir la prise de décision dans le secteur de la santé :** Fournir des informations qui peuvent aider les professionnels de santé et les patients à faire des choix éclairés concernant les médicaments.

3. Besoins analytiques

- **Suivi de la sécurité des médicaments** : Suivre la fréquence et les types d'événements indésirables signalés pour chaque médicament.

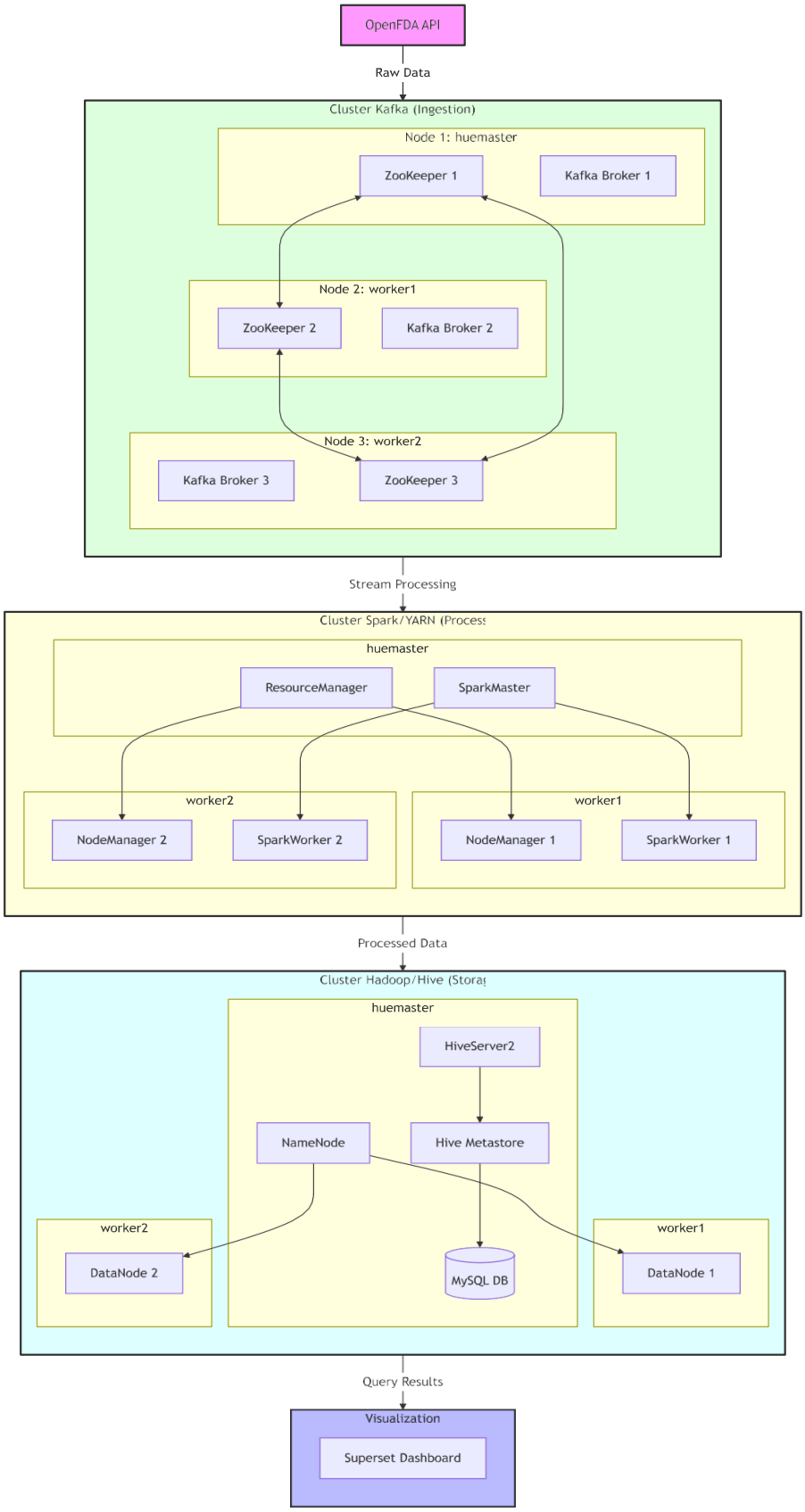
- **Profilage des risques** : Évaluer les médicaments en fonction de la gravité des effets indésirables signalés afin de déterminer ceux qui peuvent présenter des risques élevés pour la santé.

4. Types d'analyse

- **Analyse descriptive** : Examiner la fréquence et les types d'événements indésirables associés aux différents médicaments, ainsi que les tendances générales des retraits de médicaments.

- **Analyse diagnostique** : Identifier les facteurs sous-jacents ou les caractéristiques communes des médicaments fréquemment rappelés ou signalés pour des événements indésirables.

Notre Première Architecture



L'architecture mise en place pour l'analyse des risques liés aux médicaments constitue une solution Big Data robuste et scalable, intégrant plusieurs technologies open-source de pointe. Cette solution vise à collecter, traiter et analyser efficacement les données de l'API OpenFDA pour fournir des insights précieux sur la sécurité des médicaments.

Au cœur de l'architecture, nous retrouvons quatre couches principales interconnectées, chacune jouant un rôle crucial dans le pipeline de données :

La première couche d'ingestion s'appuie sur Apache Kafka, un système de messagerie distribué déployé sur trois nœuds (huemaster, worker1, worker2). Chaque nœud héberge un broker Kafka et une instance ZooKeeper, formant ainsi un cluster hautement disponible. Les brokers Kafka, assurent la persistance et la distribution fiable des données collectées depuis l'API OpenFDA.

ZooKeeper, fonctionnant en mode quorum avec trois serveurs, garantit la coordination et la gestion des métadonnées du cluster Kafka. Cette configuration permet de gérer efficacement les flux de données entrants tout en assurant la tolérance aux pannes.

La deuxième couche, dédiée au traitement des données, utilise Apache Spark s'exécutant sur YARN. Le nœud huemaster héberge le ResourceManager YARN et le SparkMaster, tandis que les workers exécutent les NodeManagers et les SparkWorkers. Cette configuration permet un traitement distribué efficace avec une gestion optimisée des ressources. Spark, avec sa configuration optimisée (4GB de mémoire par executor), traite les données en temps réel et par lots, permettant des analyses complexes et des transformations de données à grande échelle.

La troisième couche gère le stockage avec un cluster Hadoop/Hive. Le NameNode HDFS et HiveServer2 sont déployés sur huemaster, tandis que les DataNodes résident sur les workers. HDFS assure un stockage distribué avec un facteur de réplication de 3 pour une haute disponibilité des données. Hive, avec son metastore MySQL, fournit une interface SQL pour interroger les données stockées dans HDFS.

La couche finale de visualisation utilise Apache Superset, déployé sur huemaster. Superset se connecte directement à Hive via une connexion optimisée et sécurisée, offrant une interface utilisateur web intuitive pour créer des tableaux de bord interactifs et des visualisations personnalisées.

Les performances sont optimisées grâce à plusieurs mécanismes : compression des données à différents niveaux (Kafka, HDFS, Hive), parallélisation des traitements, mise en cache intelligente, et configuration fine des ressources (mémoire, CPU) pour chaque composant.

Cette architecture assure non seulement le traitement efficace des données pharmaceutiques mais garantit aussi la scalabilité horizontale, permettant d'augmenter la capacité du système en ajoutant simplement de nouveaux nœuds workers. La haute disponibilité est assurée par la redondance des composants critiques et la réplication des données, minimisant ainsi les risques d'interruption de service.

L'ensemble du système est conçu pour évoluer avec les besoins, que ce soit en termes de volume de données, de complexité des analyses ou de nombre d'utilisateurs, tout en maintenant des performances optimales et une fiabilité maximale.

Nécessité d'une Architecture Big Data pour l'Analyse des Risques Médicamenteux

L'analyse des risques médicamenteux nécessite aujourd'hui une solution Big Data pour traiter efficacement les millions de rapports d'effets indésirables générés annuellement par l'API OpenFDA. La nature critique de ces données de santé exige un traitement en temps réel pour détecter rapidement les risques potentiels et alerter les professionnels de santé sans délai.

La diversité des sources d'information, allant des données structurées aux rapports cliniques textuels, demande une infrastructure capable de gérer et d'analyser simultanément différents types de données.

Cette solution Big Data distribuée assure non seulement le traitement massif des données mais garantit aussi la haute disponibilité nécessaire pour un service de santé publique, tout en permettant une adaptation flexible aux besoins croissants d'analyse.

3. Architecture Technique

3.1 Cluster Kafka

- 3 brokers Kafka (un par nœud)
- Cluster ZooKeeper distribué
- Configuration haute disponibilité
- Réplication factor : 2
- Ports :
  - Kafka : 9092
  - ZooKeeper : 2181, 2888, 3888

3.2 Cluster Spark/YARN

- Mode déploiement : YARN
- Configuration mémoire :
  - Driver : 4GB
  - Executor : 4GB
  - Application Master : 1GB

3.3 Cluster Hadoop

- NameNode : huemaster
- DataNodes : worker1, worker2
- Réplication HDFS : 3
- Configuration YARN

3.4 Cluster Hive

- HiveServer2 sur huemaster
- Metastore MySQL sur huemaster

3.5 Superset

- Déployé sur huemaster
- Intégration avec Hive

Producer.py

```
from kafka import KafkaProducer
import json
import time
import requests
def fetch_fda_data():
	url = "https://api.fda.gov/drug/event.json"
	params = {
	"limit": 20,
	"search": "receivedate:[20240101 TO 20240331]"
	}
	try:
	response = requests.get(url, params=params)
	response.raise_for_status()
	return response.json().get("results", [])
	except Exception as e:
	print(f"Erreur lors de la récupération des données: {str(e)}")
	return []
def transform_data(result):
	return {
	"safetyreportid": result.get("safetyreportid"),
	"receivedate": result.get("receivedate"),
	"serious": result.get("serious"),
	"drugs": [
		{
		"medicinalproduct": drug.get("medicinalproduct"),
		"drugindication": drug.get("drugindication")
		}
		for drug in result.get("patient", {}).get("drug", [])
	],
	"reactions": [
```

```
            reaction.get("reactionmeddrapt")
            for reaction in result.get("patient", {}).get("reaction", [])]}
def main():
      producer = KafkaProducer(
      bootstrap_servers=['huemaster:9092'],
      value_serializer=lambda v: json.dumps(v).encode('utf-8'),
      client_id='fda_producer'
      )
      try:
      results = fetch_fda_data()
      for i, result in enumerate(results, 1):
            transformed_data = transform_data(result)
            producer.send('devoir', value=transformed_data)
            print(f"Message {i}/20 envoyé")
            time.sleep(1)
      producer.flush()
      print("Tous les messages ont été envoyés avec succès")
      except Exception as e:
      print(f"Erreur lors de l'envoi des messages: {str(e)}")
      finally:
      producer.close()
if __name__ == "__main__":
      main()
```

Consumer.py

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import from_json, col, explode
from pyspark.sql.types import StructType, StructField, StringType, ArrayType
# Fonction pour définir le schéma des messages
def get_schema():
      return StructType([
      StructField("safetyreportid", StringType(), True),
      StructField("receivedate", StringType(), True),
```

```
        StructField("serious", StringType(), True),  # Champ corrigé
        StructField("drugs", ArrayType(StructType([
                StructField("medicinalproduct", StringType(), True),
                StructField("drugindication", StringType(), True)
        ])), True),
        StructField("reactions", ArrayType(StringType()), True)  # Champ ajouté
        ])
# Fonction pour créer une session Spark
def create_spark_session():
        return SparkSession.builder \
        .appName("OpenFDA_Streaming_20") \
        .config("spark.jars.packages", "org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.8") \
        .config("spark.sql.warehouse.dir", "hdfs://huemaster:9000/user/hive/warehouse") \
        .config("spark.kafka.consumer.group.id", "medical_data_group") \
        .config("hive.metastore.uris", "thrift://huemaster:9083") \
        .enableHiveSupport() \
        .getOrCreate()
message_count = 0
MAX_MESSAGES = 20
# Fonction pour traiter chaque batch de données
def process_batch(df, epoch_id):
        global message_count
        try:
        if message_count >= MAX_MESSAGES:
                print(f"Message count reached {MAX_MESSAGES}, stopping processing.")
                return False
        batch_count = df.count()
        if batch_count == 0:
                return True
        print(f"\nTraitement batch {epoch_id} - {batch_count} messages")
        drugs_df = df.select(
                col("safetyreportid"),
                col("receivedate"),
                (col("serious") == "1").alias("serious"),  # Conversion en booléen
                explode("drugs").alias("drug")
```

```
        ).select(
                "safetyreportid",
                "receivedate",
                "serious",
                "drug.medicinalproduct",
                "drug.drugindication"
        )
        remaining = MAX_MESSAGES - message_count
        if batch_count > remaining:
                drugs_df = drugs_df.limit(remaining)
                batch_count = remaining
        print("\nDonnées reçues :")
        drugs_df.show(5, truncate=False)
        drugs_df.write.insertInto("medical_data.adverse_events", overwrite=False)
        message_count += batch_count
        print(f"Total traité : {message_count}/{MAX_MESSAGES}")
        print("\nContenu actuel de la table:")
        spark.sql("SELECT * FROM medical_data.adverse_events").show()
        return message_count < MAX_MESSAGES
        except Exception as e:
        print(f"Erreur batch {epoch_id}: {str(e)}")
        return True
# Fonction pour initialiser les tables Hive
def init_tables(spark):
        print("Initialisation de la base de données...")
        spark.sql("DROP DATABASE IF EXISTS medical_data CASCADE")
        spark.sql("CREATE DATABASE medical_data")
        create_table_sql = """
        CREATE TABLE medical_data.adverse_events (
        safetyreportid STRING,
        receivedate STRING,
        serious BOOLEAN,
        medicinalproduct STRING,
        drugindication STRING
        ) USING PARQUET
```

```
        spark.sql(create_table_sql)
        print("Tables créées avec succès")
def main():
        try:
        global spark
        spark = create_spark_session()
        spark.sparkContext.setLogLevel("WARN")
        init_tables(spark)
        print("Configuration Kafka...")
        df = spark.readStream \
        .format("kafka") \
        .option("kafka.bootstrap.servers", "huemaster:9092") \
        .option("subscribe", "devoir") \
        .option("startingOffsets", "earliest") \
        .option("maxOffsetsPerTrigger", 5) \
        .option("failOnDataLoss", "false") \
        .load()
        parsed_df = df.selectExpr("CAST(value AS STRING)") \
        .select(from_json(col("value"), get_schema()).alias("data")) \
        .select("data.*")
        print("Démarrage streaming...")
        query = parsed_df \
        .writeStream \
        .foreachBatch(process_batch) \
        .option("checkpointLocation", "/tmp/checkpoint/devoir") \
        .trigger(processingTime='5 seconds') \
        .start()
        query.awaitTermination()
        if query.isActive:
        print("\nArrêt du streaming...")
        query.stop()
        print("\nRésultat final:")
        spark.sql("SELECT * FROM medical_data.adverse_events").show()
        except Exception as e:
        print(f"Erreur principale : {str(e)}")
        finally:
        if 'spark' in locals():
```

```
        spark.stop()

if __name__ == "__main__":

        main()
```

4. Configuration Détaillée

Configuration Kafka Distribuée sur 3 Machines

Machine 1 (huemaster)

server.properties

```
# Identifiant unique du broker
broker.id=1
# Listeners pour les connexions clients
listeners=PLAINTEXT://huemaster:9092
advertised.listeners=PLAINTEXT://huemaster:9092
# Configuration du stockage
log.dirs=/opt/kafka/data/broker1
# Configuration de base
num.partitions=3
default.replication.factor=2
min.insync.replicas=2

# Connexion ZooKeeper
zookeeper.connect=huemaster:2181,worker1:2181,worker2:2181
# Configuration du leader
auto.leader.rebalance.enable=true
leader.imbalance.check.interval.seconds=300

# Configuration de performance
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
num.recovery.threads.per.data.dir=1

# Configuration de rétention
log.retention.hours=168
log.retention.bytes=1073741824
log.segment.bytes=1073741824
log.retention.check.interval.ms=300000

# Configuration de transaction
transaction.state.log.replication.factor=2
transaction.state.log.min.isr=2
```

zookeeper.properties

```
# Configuration de base
dataDir=/opt/kafka/zookeeper/data
dataLogDir=/opt/kafka/zookeeper/logs
clientPort=2181
maxClientCnxns=0
```

```
admin.enableServer=true
admin.serverPort=8080


# Configuration du cluster
initLimit=5
syncLimit=2
tickTime=2000

# Définition des serveurs
server.1=huemaster:2888:3888
server.2=worker1:2888:3888
server.3=worker2:2888:3888

# Configuration de sécurité
authProvider.1=org.apache.zookeeper.server.auth.SASLAuthenticationProvider
requireClientAuthScheme=sasl
```

myid

```
1
```

Machine 2 (worker1)

server.properties

```
broker.id=2
listeners=PLAINTEXT://worker1:9092
advertised.listeners=PLAINTEXT://worker1:9092
log.dirs=/opt/kafka/data/broker2

# Mêmes configurations que broker 1
num.partitions=3
default.replication.factor=2
min.insync.replicas=2
zookeeper.connect=huemaster:2181,worker1:2181,worker2:2181

# Configuration de performance
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
```

zookeeper.properties

```
dataDir=/opt/kafka/zookeeper/data
dataLogDir=/opt/kafka/zookeeper/logs
clientPort=2181
maxClientCnxns=0
admin.enableServer=true
admin.serverPort=8080

initLimit=5
syncLimit=2
tickTime=2000
```

```
server.1=huemaster:2888:3888
server.2=worker1:2888:3888
server.3=worker2:2888:3888
```

myid

```
2
```

Machine 3 (worker2)

server.properties

```
broker.id=3
listeners=PLAINTEXT://worker2:9092
advertised.listeners=PLAINTEXT://worker2:9092
log.dirs=/opt/kafka/data/broker3

# Mêmes configurations que broker 1 et 2
num.partitions=3
default.replication.factor=2
min.insync.replicas=2
zookeeper.connect=huemaster:2181,worker1:2181,worker2:2181

# Configuration de performance
num.network.threads=3
num.io.threads=8
socket.send.buffer.bytes=102400
socket.receive.buffer.bytes=102400
socket.request.max.bytes=104857600
```

zookeeper.properties

```
dataDir=/opt/kafka/zookeeper/data
dataLogDir=/opt/kafka/zookeeper/logs
clientPort=2181
maxClientCnxns=0
admin.enableServer=true
admin.serverPort=8080

initLimit=5
syncLimit=2
tickTime=2000

server.1=huemaster:2888:3888
server.2=worker1:2888:3888
server.3=worker2:2888:3888
```

myid

```
3
```

Scripts de Démarrage

start-kafka.sh

```
#!/bin/bash
# Démarrage de ZooKeeper
/opt/kafka/bin/zookeeper-server-start.sh -daemon
/opt/kafka/config/zookeeper.properties
```

```
# Attente de 10 secondes pour que ZooKeeper démarre
sleep 10
# Démarrage du broker Kafka
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
```

stop-kafka.sh

```
#!/bin/bash
# Arrêt du broker Kafka
/opt/kafka/bin/kafka-server-stop.sh
# Attente de 5 secondes
sleep 5
# Arrêt de ZooKeeper
/opt/kafka/bin/zookeeper-server-stop.sh
```

Commandes de supervision

```
# Vérification des topics
/opt/kafka/bin/kafka-topics.sh --bootstrap-server huemaster:9092 --list
# Vérification des groupes de consommateurs
/opt/kafka/bin/kafka-consumer-groups.sh --bootstrap-server huemaster:9092 --list
# Vérification du statut du cluster
/opt/kafka/bin/kafka-topics.sh --describe --bootstrap-server huemaster:9092
```

Configuration Spark Distribuée

Master Node (huemaster)

spark-env.sh

```
#!/usr/bin/env bash
# Configuration des variables d'environnement Java
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export SPARK_DIST_CLASSPATH=$(hadoop classpath)


# Configuration Spark Master
export SPARK_MASTER_HOST=huemaster
export SPARK_MASTER_PORT=7077
export SPARK_MASTER_WEBUI_PORT=8080

# Configuration de la mémoire
export SPARK_MASTER_OPTS="-Xmx4g -XX:+UseG1GC -XX:+CMSClassUnloadingEnabled"

# Configuration des logs
export SPARK_LOG_DIR=/var/log/spark
export SPARK_WORKER_DIR=/var/lib/spark

# Configuration pour Hadoop/YARN
export HADOOP_CONF_DIR=/etc/hadoop/conf
export YARN_CONF_DIR=/etc/hadoop/conf

# Configuration Python
export PYSPARK_PYTHON=/usr/bin/python3
export PYSPARK_DRIVER_PYTHON=/usr/bin/python3
```

spark-defaults.conf

```
# Configuration générale
spark.master                     yarn
spark.driver.memory              4g
spark.executor.memory            4g
spark.executor.instances         2
spark.executor.cores             4
spark.default.parallelism        16

# Configuration du mode de déploiement
spark.submit.deployMode          cluster
spark.driver.cores               2

# Configuration de la mémoire
spark.memory.fraction            0.8
spark.memory.storageFraction     0.3
spark.memory.offHeap.enabled     true
spark.memory.offHeap.size        2g

# Configuration de la sérialisation
spark.serializer                 org.apache.spark.serializer.KryoSerializer
spark.kryoserializer.buffer.max   1g
spark.rdd.compress               true

# Configuration du shuffle
spark.shuffle.file.buffer        32k
spark.shuffle.compress           true
spark.shuffle.spill.compress     true
spark.shuffle.service.enabled    true

# Configuration des événements et de l'historique
spark.eventLog.enabled           true
spark.eventLog.dir               hdfs://huemaster:9000/spark-logs
spark.history.provider           org.apache.spark.deploy.history.FsHistoryProvider
spark.history.fs.logDirectory hdfs://huemaster:9000/spark-logs
spark.history.fs.update.interval  10s
spark.history.ui.port            18080

# Configuration des métriques
spark.metrics.conf.*.sink.graphite.class=org.apache.spark.metrics.sink.GraphiteSink
spark.metrics.conf.*.sink.graphite.host=monitoring.example.com
spark.metrics.conf.*.sink.graphite.port=2003
spark.metrics.conf.*.sink.graphite.period=10
spark.metrics.conf.*.source.jvm.class=org.apache.spark.metrics.source.JvmSource

# Configuration de l'intégration
spark.sql.warehouse.dir       hdfs://huemaster:9000/user/hive/warehouse
spark.hadoop.fs.defaultFS    hdfs://huemaster:9000
```

Worker Nodes (worker1, worker2)

spark-env.sh

```
#!/usr/bin/env bash
```

```
# Configuration des variables d'environnement Java
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export SPARK_DIST_CLASSPATH=$(hadoop classpath)

# Configuration Worker
export SPARK_WORKER_CORES=8
export SPARK_WORKER_MEMORY=8g
export SPARK_WORKER_PORT=7078
export SPARK_WORKER_WEBUI_PORT=8081

# Configuration des logs
export SPARK_LOG_DIR=/var/log/spark
export SPARK_WORKER_DIR=/var/lib/spark

# Configuration pour Hadoop/YARN
export HADOOP_CONF_DIR=/etc/hadoop/conf
export YARN_CONF_DIR=/etc/hadoop/conf

# Configuration des métriques workers
export SPARK_WORKER_OPTS="-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=8100 -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false"
```

Scripts de Démarrage et d'Arrêt

start-cluster.sh

```
#!/bin/bash
# Démarrage du master
/opt/spark/sbin/start-master.sh

# Attente du démarrage du master
sleep 5

# Démarrage des workers
ssh worker1 "/opt/spark/sbin/start-worker.sh spark://huemaster:7077"
ssh worker2 "/opt/spark/sbin/start-worker.sh spark://huemaster:7077"
# Démarrage de l'historique
/opt/spark/sbin/start-history-server.sh
```

stop-cluster.sh

```
#!/bin/bash
# Arrêt des workers
ssh worker1 "/opt/spark/sbin/stop-worker.sh"
ssh worker2 "/opt/spark/sbin/stop-worker.sh"

# Arrêt du master
/opt/spark/sbin/stop-master.sh

# Arrêt de l'historique
/opt/spark/sbin/stop-history-server.sh
```

Configuration Yarn-Site.xml pour Spark

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>spark_shuffle,mapreduce_shuffle</value>
    </property>
    <property>


        <name>yarn.nodemanager.aux-services.spark_shuffle.class</name>
        <value>org.apache.spark.network.yarn.YarnShuffleService</value>
    </property>

    <property>
        <name>yarn.scheduler.minimum-allocation-mb</name>
        <value>1024</value>
    </property>

    <property>
        <name>yarn.scheduler.maximum-allocation-mb</name>
        <value>8192</value>
    </property>
</configuration>
```

Commandes de Supervision

```
# Vérification du statut du cluster
/opt/spark/bin/spark-submit --status
# Liste des applications
yarn application -list
```

Configuration Hadoop Distribuée

Configuration Environnement (/etc/profile.d/hadoop.sh)

```
# Variables d'environnement Hadoop
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
# Configuration Java
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
```

Configuration Principale (common)

core-site.xml

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://huemaster:9000</value>
```

```
    </property>

    <property>
        <name>io.file.buffer.size</name>
        <value>131072</value>
    </property>

    <property>
        <name>hadoop.tmp.dir</name>
        <value>/data/hadoop/tmp</value>
    </property>

    <property>
        <name>hadoop.http.staticuser.user</name>
        <value>hdfs</value>
    </property>

    <property>
        <name>ha.zookeeper.quorum</name>
        <value>huemaster:2181,worker1:2181,worker2:2181</value>
    </property>
</configuration>
```

hdfs-site.xml

```
<configuration>
    <!-- Configuration NameNode -->
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>/data/hadoop/namenode</value>
    </property>

    <property>
        <name>dfs.namenode.handler.count</name>
        <value>100</value>
    </property>

    <!-- Configuration DataNode -->
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/data/hadoop/datanode</value>
    </property>

    <property>
        <name>dfs.datanode.handler.count</name>
        <value>20</value>
    </property>

    <!-- Configuration Réplication -->
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>

    <property>
        <name>dfs.blocksize</name>
```

```
        <value>134217728</value>
    </property>

    <!-- Configuration Permission -->
    <property>
        <name>dfs.permissions.enabled</name>
        <value>true</value>
    </property>

    <property>
        <name>dfs.permissions.superusergroup</name>
        <value>hadoop</value>
    </property>

    <!-- Configuration WebHDFS -->
    <property>
        <name>dfs.webhdfs.enabled</name>
        <value>true</value>
    </property>

    <!-- Configuration HA -->
    <property>
        <name>dfs.nameservices</name>
        <value>hadoop-cluster</value>
    </property>
</configuration>
```

yarn-site.xml

```
<configuration>
    <!-- Configuration ResourceManager -->
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>huemaster</value>
    </property>

    <property>
        <name>yarn.resourcemanager.webapp.address</name>
        <value>huemaster:8088</value>
    </property>

    <!-- Configuration NodeManager -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>

    <property>
        <name>yarn.nodemanager.resource.memory-mb</name>
        <value>16384</value>
    </property>

    <property>
        <name>yarn.nodemanager.resource.cpu-vcores</name>
        <value>8</value>
    </property>
```

```
    <!-- Configuration Scheduler -->
    <property>
        <name>yarn.scheduler.minimum-allocation-mb</name>
        <value>1024</value>
    </property>

    <property>
        <name>yarn.scheduler.maximum-allocation-mb</name>
        <value>8192</value>
    </property>

    <!-- Configuration Log Aggregation -->
    <property>
        <name>yarn.log-aggregation-enable</name>
        <value>true</value>
    </property>

    <property>
        <name>yarn.log-aggregation.retain-seconds</name>
        <value>604800</value>
    </property>
</configuration>
```

mapred-site.xml

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>

    <!-- Configuration MapReduce Job -->
    <property>
        <name>mapreduce.map.memory.mb</name>
        <value>2048</value>
    </property>

    <property>
        <name>mapreduce.reduce.memory.mb</name>
        <value>4096</value>
    </property>

    <property>
        <name>mapreduce.map.java.opts</name>
        <value>-Xmx1638m</value>
    </property>

    <property>
        <name>mapreduce.reduce.java.opts</name>
        <value>-Xmx3278m</value>
    </property>

    <!-- Configuration Histoire -->
    <property>
        <name>mapreduce.jobhistory.address</name>
```

```
        <value>huemaster:10020</value>
    </property>

    <property>
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>huemaster:19888</value>
    </property>
</configuration>
```

Scripts de Démarrage/Arrêt

start-hadoop.sh

```
#!/bin/bash
# Formatage du NameNode (première fois uniquement)
# hdfs namenode -format

# Démarrage HDFS
start-dfs.sh


sleep 10

# Démarrage YARN
start-yarn.sh

# Démarrage de l'historique des jobs
mr-jobhistory-daemon.sh start historyserver

# Vérification de l'état du cluster
hdfs dfsadmin -report
yarn node -list
```

stop-hadoop.sh

```
#!/bin/bash

# Arrêt de l'historique des jobs
mr-jobhistory-daemon.sh stop historyserver

# Arrêt YARN
stop-yarn.sh

# Arrêt HDFS
stop-dfs.sh
```

Configuration des Workers

workers

```
worker1
worker2
```

Commandes de Maintenance

```
# Vérification de l'état du système de fichiers
hdfs fsck /
# Rapport sur l'utilisation du système de fichiers
```

```
hdfs dfs -df -h
# Liste des nœuds YARN
yarn node -list -all
# Vérification des applications YARN
yarn application -list
# Vérification de l'état des DataNodes
hdfs dfsadmin -report
```

Configuration Apache Hive Distribuée

Configuration Principale (hive-site.xml)

```
<configuration>
    <!-- Configuration MySQL Metastore -->
    <property>
        <name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://huemaster:3306/metastore?createDatabaseIfNotExist=true&amp;useSSL
=false&amp;allowPublicKeyRetrieval=true</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>com.mysql.cj.jdbc.Driver</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionUserName</name>
        <value>hive</value>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionPassword</name>
        <value>hive_password</value>
    </property>

    <!-- Configuration Metastore -->
    <property>
        <name>hive.metastore.warehouse.dir</name>
        <value>/user/hive/warehouse</value>
    </property>
    <property>
        <name>hive.metastore.uris</name>
        <value>thrift://huemaster:9083</value>
    </property>
    <property>
        <name>hive.metastore.event.db.notification.api.auth</name>
        <value>false</value>
    </property>
    <property>
        <name>hive.metastore.client.socket.timeout</name>
        <value>3600</value>
    </property>
    <property>
        <name>hive.metastore.warehouse.external.dir</name>
        <value>/user/hive/external</value>
    </property>
```

```
<!-- Configuration HiveServer2 -->
<property>
    <name>hive.server2.thrift.bind.host</name>
    <value>huemaster</value>
</property>
<property>
    <name>hive.server2.thrift.port</name>
    <value>10000</value>
</property>
<property>
    <name>hive.server2.transport.mode</name>
    <value>binary</value>
</property>
<property>
    <name>hive.server2.authentication</name>
    <value>NONE</value>
</property>
<property>
    <name>hive.server2.enable.doAs</name>
    <value>false</value>
</property>
<property>
    <name>hive.server2.thrift.min.worker.threads</name>
    <value>5</value>
</property>
<property>
    <name>hive.server2.thrift.max.worker.threads</name>
    <value>500</value>
</property>

<!-- Configuration de Performance -->
<property>
    <name>hive.optimize.reducededuplication</name>
    <value>true</value>
</property>
<property>
    <name>hive.optimize.reducededuplication.min.reducer</name>
    <value>4</value>
</property>
<property>
    <name>hive.exec.parallel</name>
    <value>true</value>
</property>
<property>
    <name>hive.exec.parallel.thread.number</name>
    <value>8</value>
</property>
<property>
    <name>hive.exec.max.dynamic.partitions</name>
    <value>5000</value>
</property>
<property>
    <name>hive.exec.max.dynamic.partitions.pernode</name>
    <value>2000</value>
```

```
    </property>
    <property>
        <name>hive.auto.convert.join</name>
        <value>true</value>
    </property>
    <property>
        <name>hive.auto.convert.join.noconditionaltask.size</name>
        <value>10000000</value>
    </property>

    <!-- Configuration des Statistics -->
    <property>
        <name>hive.stats.autogather</name>
        <value>true</value>
    </property>
    <property>
        <name>hive.stats.fetch.column.stats</name>
        <value>true</value>
    </property>
    <property>
        <name>hive.compute.query.using.stats</name>
        <value>true</value>
    </property>

    <!-- Configuration du Cache -->
    <property>
        <name>hive.cache.expr.evaluation</name>
        <value>true</value>
    </property>
    <property>
        <name>hive.metastore.cache.pinobjtypes</name>
        <value>Table,Database,Type,FieldSchema,Order</value>
    </property>

    <!-- Configuration Tez -->
    <property>
        <name>hive.execution.engine</name>
        <value>tez</value>
    </property>
    <property>
        <name>hive.tez.container.size</name>
        <value>4096</value>
    </property>
    <property>
        <name>hive.tez.java.opts</name>
        <value>-Xmx3276m</value>
    </property>

    <!-- Configuration de Compression -->
    <property>
        <name>hive.exec.compress.output</name>
        <value>true</value>
    </property>
    <property>
        <name>hive.exec.compress.intermediate</name>
```

```
        <value>true</value>
    </property>
</configuration>
```

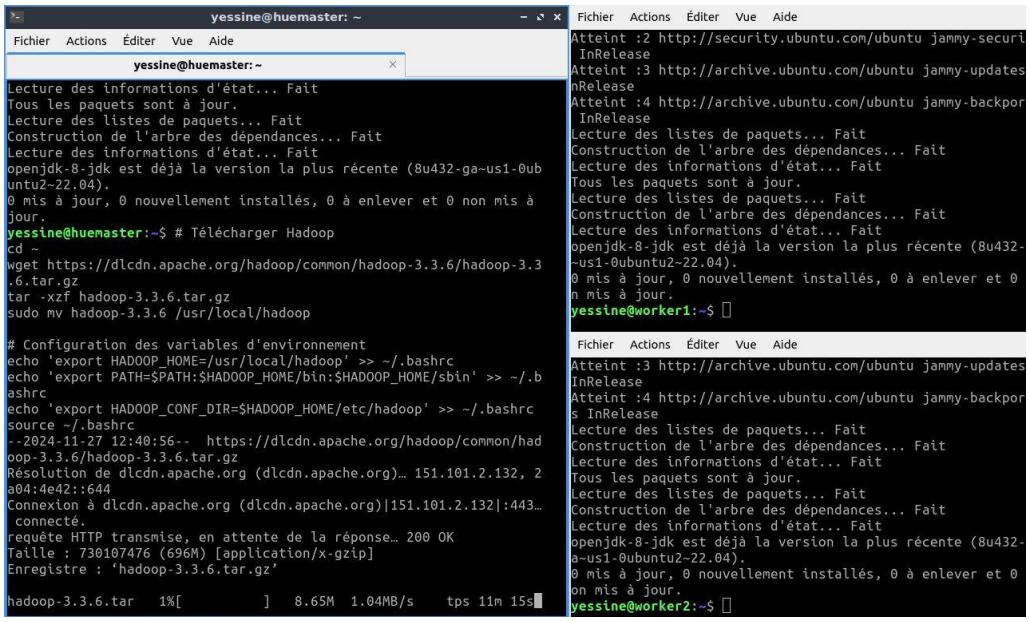
Scripts de Démarrage/Arrêt

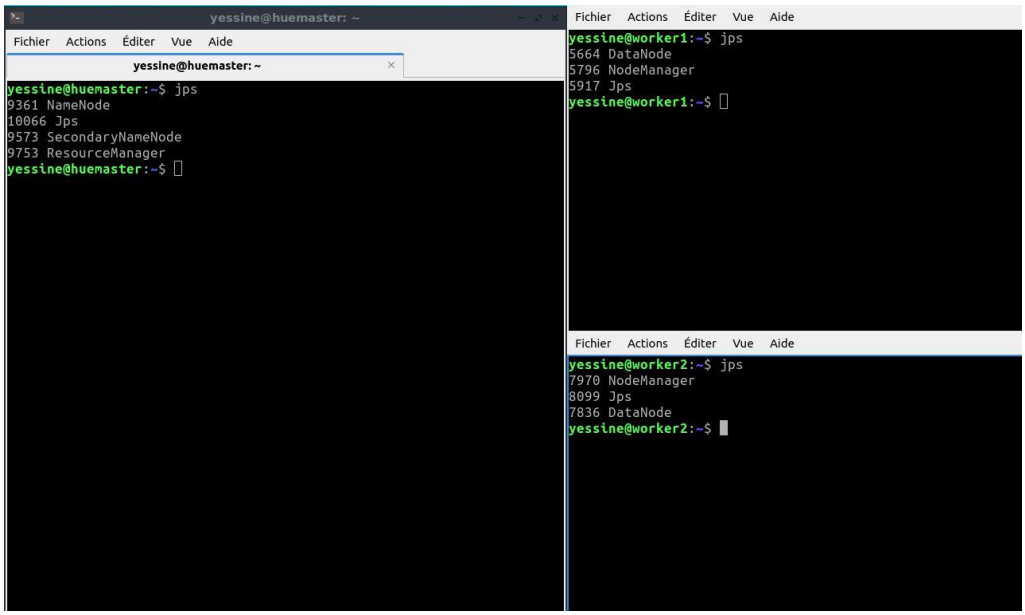
start-hive.sh

```
#!/bin/bash
# Démarrage du Metastore
nohup hive --service metastore > /var/log/hive/metastore.log 2>&1 &

# Attente du démarrage du Metastore
sleep 10

# Démarrage de HiveServer2
nohup hive --service hiveserver2 > /var/log/hive/hiveserver2.log 2>&1 &

# Vérification des processus
ps aux | grep hive
```

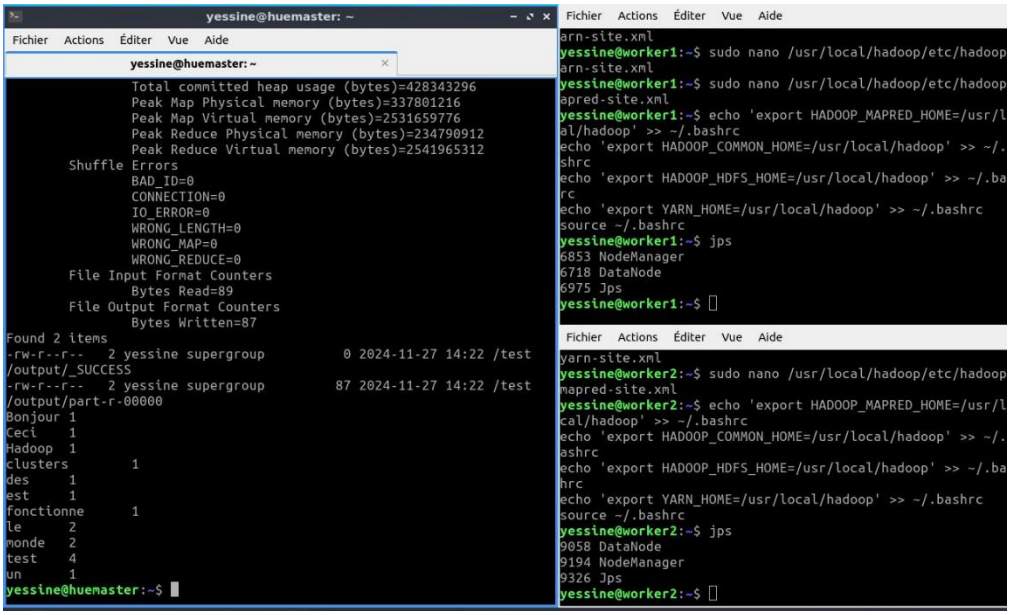
Illustration Pratique

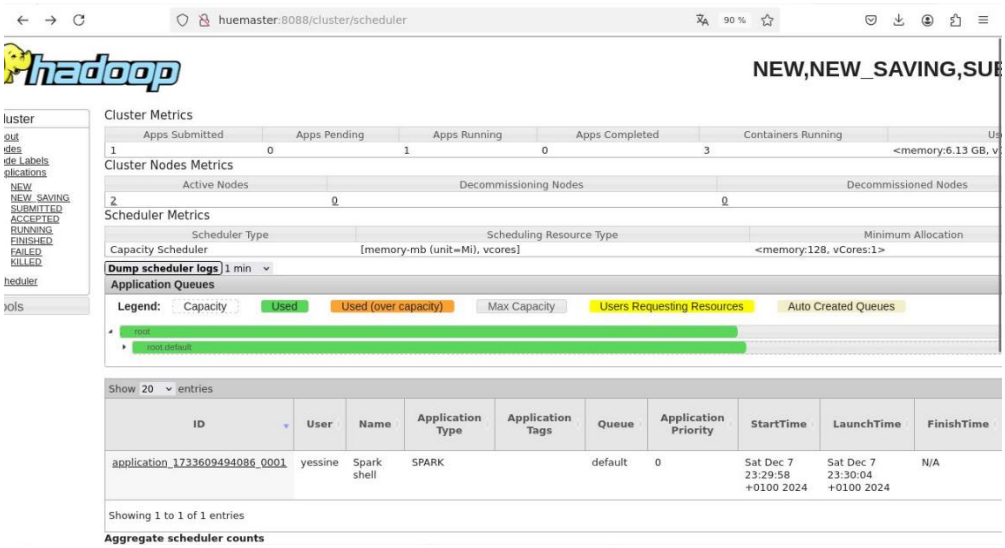
Hadoop



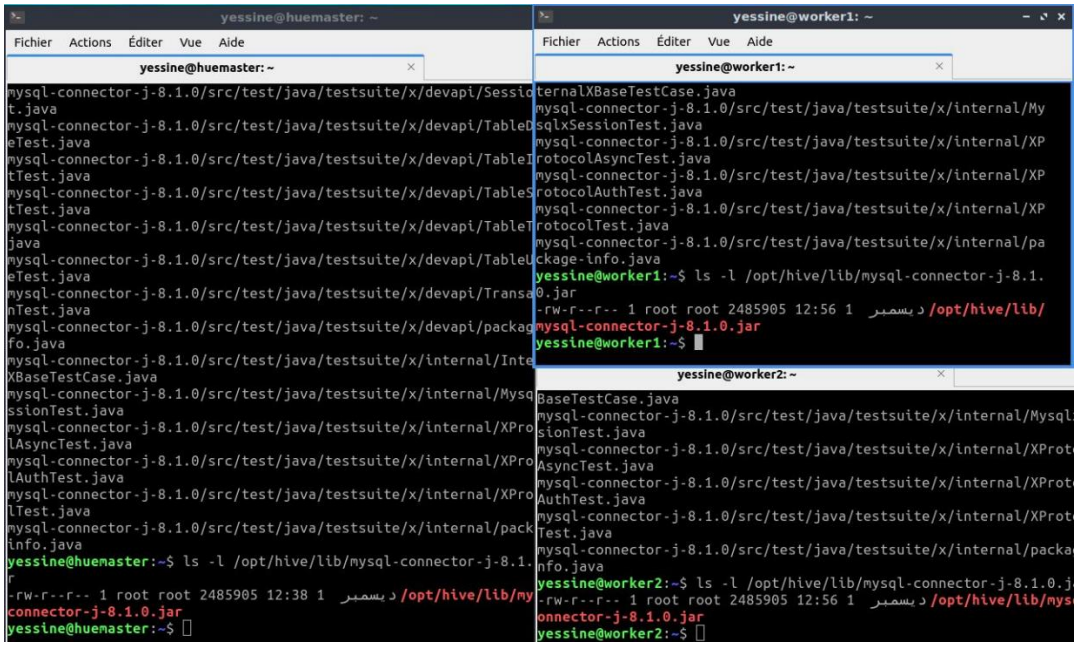


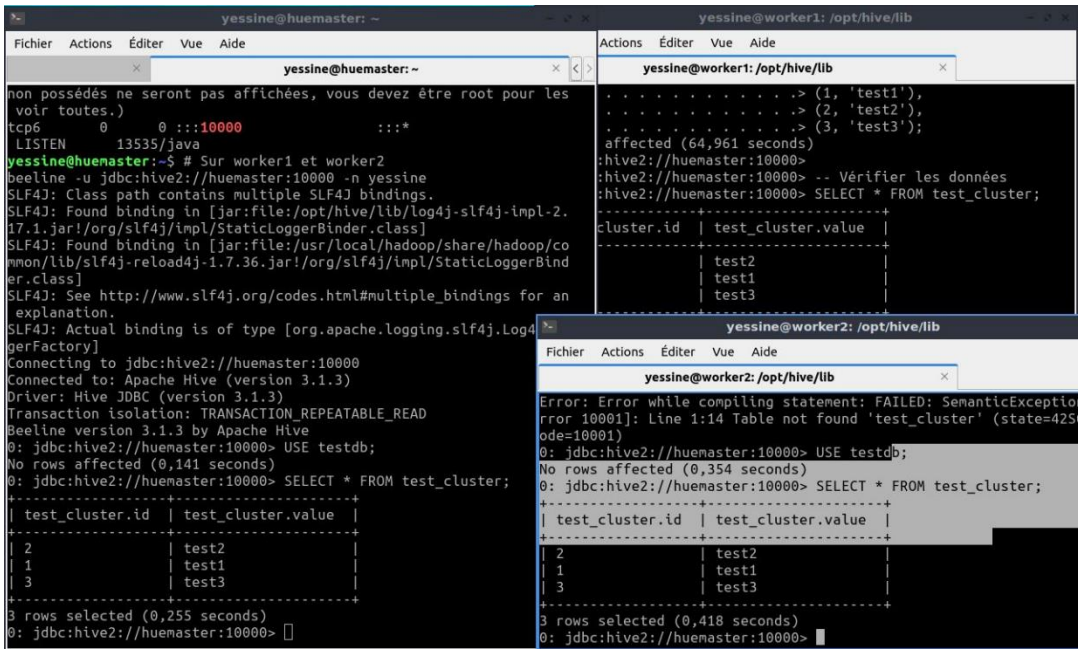
Il fonctionne bien

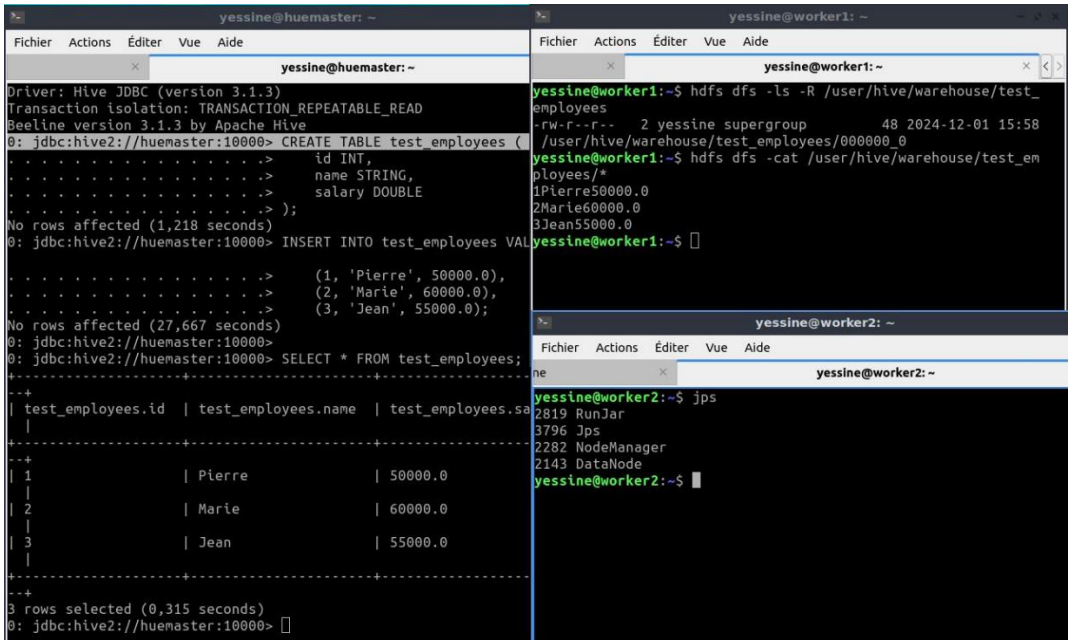



huemaster:8088/cluster/scheduler
NEW,NEW_SAVING,SU
Cluster Metrics
Apps Submitted
Apps Pending
Apps Running
Apps Completed
Containers Running
Cluster Nodes Metrics
Active Nodes
Decommissioning Nodes
Decommissioned Nodes
Scheduler Metrics
Scheduler Type
Scheduling Resource Type
Minimum Allocation
Capacity Scheduler
[memory-mb (unit=Mi), vcores]
<memory:128, vCores:1>
Application Queues
Legend:
Capacity
Used
Used (over capacity)
Max Capacity
Users Requesting Resources
Auto Created Queues
root
root.default
ID
User
Name
Application Type
Application Tags
Queue
Application Priority
StartTime
LaunchTime
FinishTime
application_1733609494086_0001
yessine
Spark shell
SPARK
default
0
Sat Dec 7 23:29:58 +0100 2024
Sat Dec 7 23:30:04 +0100 2024
N/A
Showing 1 to 1 of 1 entries
Aggregate scheduler counts

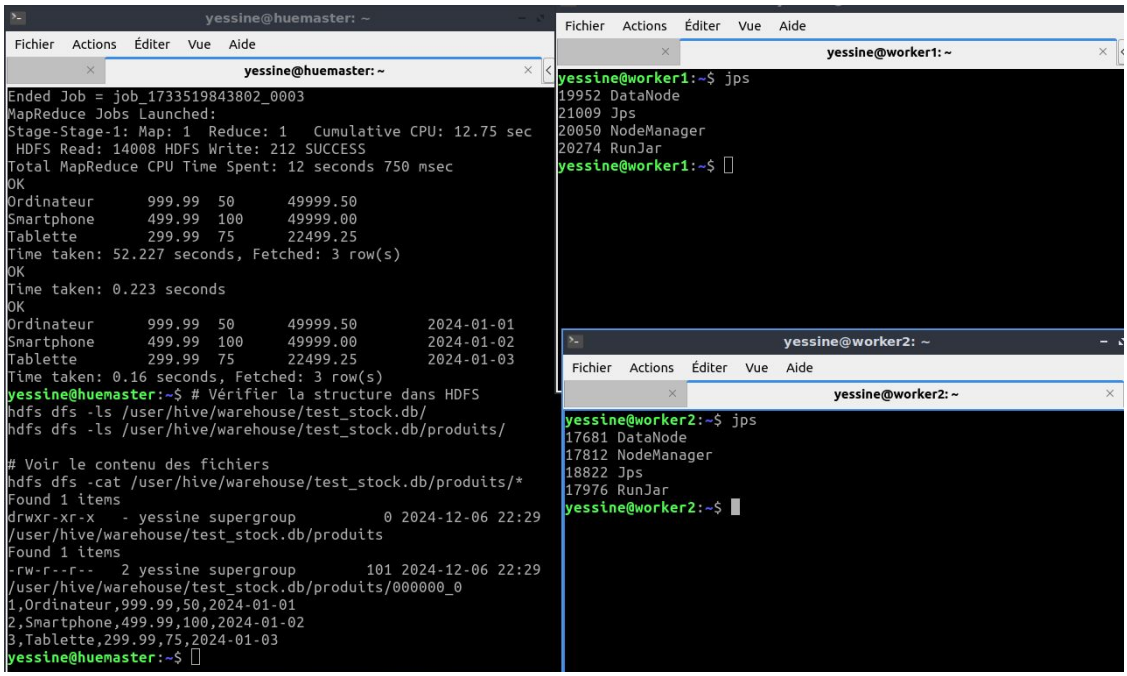
Hive


yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yessine@huemaster: ~
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/Sessio
t.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/TableD
eTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/TableI
tTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/TableS
tTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/TableT
java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/TableU
eTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/Transa
nTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/devapi/packag
fo.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/Inte
XBaseTestCase.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/Mysq
ssionTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XPro
lAsyncTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XPro
lAuthTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XPro
lTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/pack
info.java
yessine@huemaster:~$ ls -l /opt/hive/lib/mysql-connector-j-8.1.
r
-rw-r--r-- 1 root root 2485905 12:38 1 ديسمبر /opt/hive/lib/my
connector-j-8.1.0.jar
yessine@huemaster:~$
yessine@worker1: ~
Fichier Actions Éditer Vue Aide
yessine@worker1: ~
ternalXBaseTestCase.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/My
sqlxSessionTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XP
rotocolAsyncTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XP
rotocolAuthTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XP
rotocolTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/pa
ckage-info.java
yessine@worker1:~$ ls -l /opt/hive/lib/mysql-connector-j-8.1.
0.jar
-rw-r--r-- 1 root root 2485905 12:56 1 ديسمبر /opt/hive/lib/
mysql-connector-j-8.1.0.jar
yessine@worker1:~$
yessine@worker2: ~
BaseTestCase.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/Mysql
sionTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XProt
AsyncTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XProt
AuthTest.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/XProt
Test.java
mysql-connector-j-8.1.0/src/test/java/testsuite/x/internal/packa
nfo.java
yessine@worker2:~$ ls -l /opt/hive/lib/mysql-connector-j-8.1.0.j
-rw-r--r-- 1 root root 2485905 12:56 1 ديسمبر /opt/hive/lib/mys
onnector-j-8.1.0.jar
yessine@worker2:~$

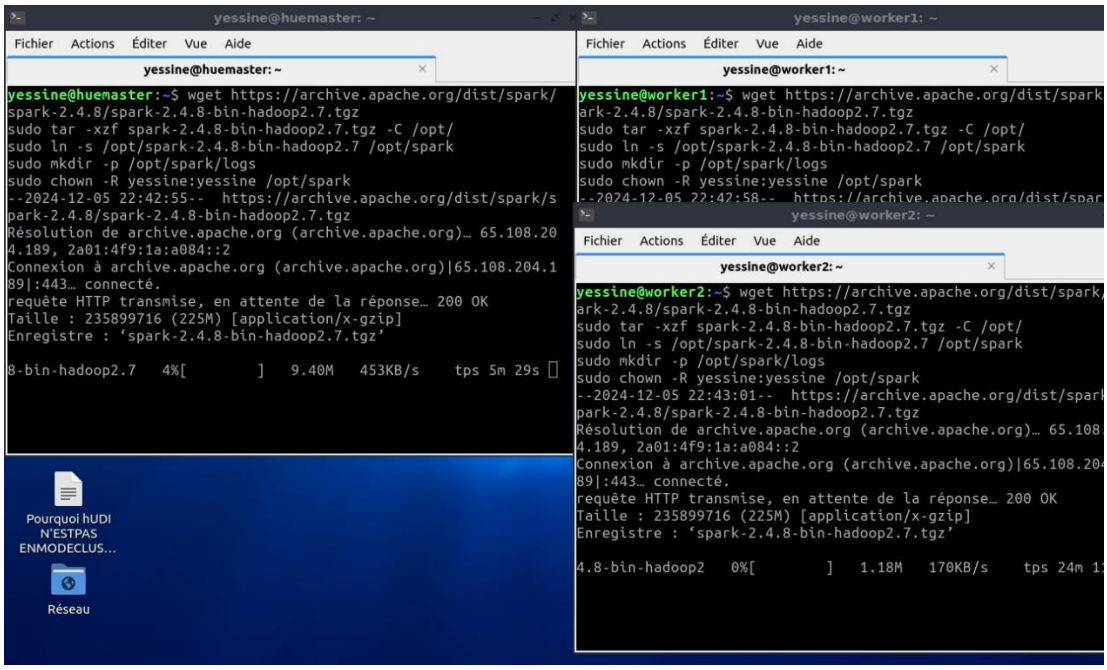

yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yessine@huemaster: ~
non possédés ne seront pas affichées, vous devez être root pour les
voir toutes.)
tcp6 0 0 :::10000 :::*
LISTEN 13535/java
yessine@huemaster:~$ # Sur worker1 et worker2
beeline -u jdbc:hive2://huemaster:10000 -n yessine
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://huemaster:10000
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://huemaster:10000> USE testdb;
No rows affected (0,141 seconds)
0: jdbc:hive2://huemaster:10000> SELECT * FROM test_cluster;
test_cluster.id	test_cluster.value
2	test2
1	test1
3	test3
3 rows selected (0,255 seconds)
0: jdbc:hive2://huemaster:10000>
yessine@worker1: /opt/hive/lib
Actions Éditer Vue Aide
yessine@worker1: /opt/hive/lib
. . . . . .> (1, 'test1'),
. . . . . .> (2, 'test2'),
. . . . . .> (3, 'test3');
affected (64,961 seconds)
:hive2://huemaster:10000>
:hive2://huemaster:10000> -- Vérifier les données
:hive2://huemaster:10000> SELECT * FROM test_cluster;
cluster.id | test_cluster.value
| test2
| test1
| test3
yessine@worker2: /opt/hive/lib
Fichier Actions Éditer Vue Aide
yessine@worker2: /opt/hive/lib
Error: Error while compiling statement: FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'test_cluster' (state=42S02,code=10001)
0: jdbc:hive2://huemaster:10000> USE testdb;
No rows affected (0,354 seconds)
0: jdbc:hive2://huemaster:10000> SELECT * FROM test_cluster;
test_cluster.id	test_cluster.value
2	test2
1	test1
3	test3
3 rows selected (0,418 seconds)
0: jdbc:hive2://huemaster:10000>

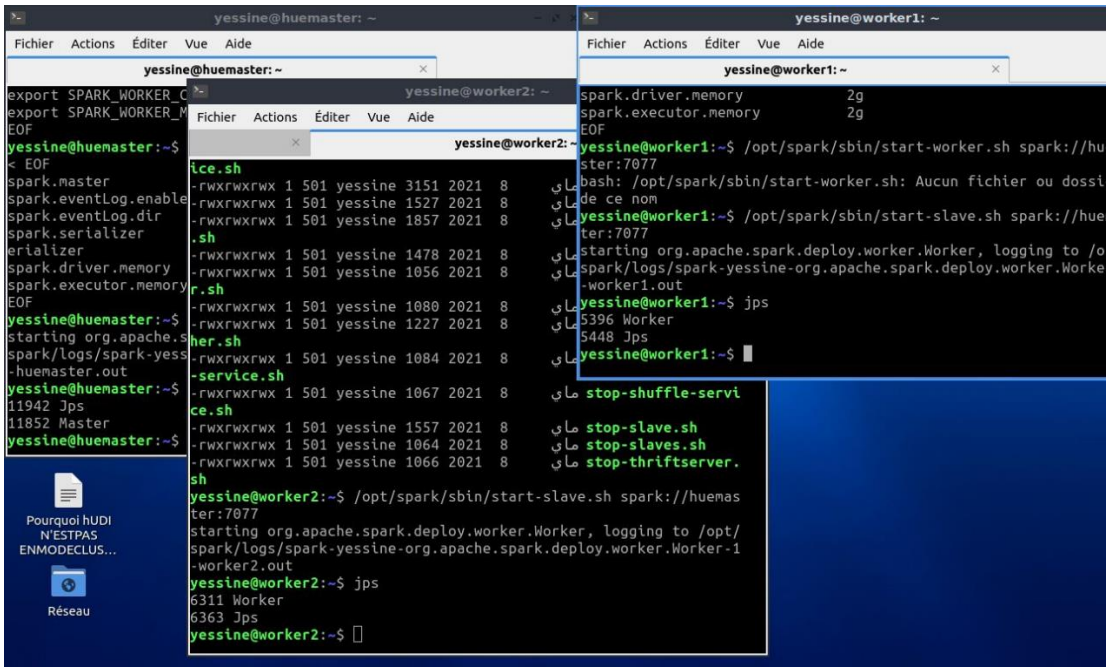

yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yessine@huemaster: ~
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://huemaster:10000> CREATE TABLE test_employees (
. . . . . .> id INT,
. . . . . .> name STRING,
. . . . . .> salary DOUBLE
. . . . . .>);
No rows affected (1,218 seconds)
0: jdbc:hive2://huemaster:10000> INSERT INTO test_employees VALUES
. . . . . .> (1, 'Pierre', 50000.0),
. . . . . .> (2, 'Marie', 60000.0),
. . . . . .> (3, 'Jean', 55000.0);
No rows affected (27,667 seconds)
0: jdbc:hive2://huemaster:10000>
0: jdbc:hive2://huemaster:10000> SELECT * FROM test_employees;
test_employees.id	test_employees.name	test_employees.salary
1	Pierre	50000.0
2	Marie	60000.0
3	Jean	55000.0
3 rows selected (0,315 seconds)
0: jdbc:hive2://huemaster:10000>
yessine@worker1: ~
Fichier Actions Éditer Vue Aide
yessine@worker1: ~
yessine@worker1:~$ hdfs dfs -ls -R /user/hive/warehouse/test_employees
-rw-r--r-- 2 yessine supergroup 48 2024-12-01 15:58 /user/hive/warehouse/test_employees/000000_0
yessine@worker1:~$ hdfs dfs -cat /user/hive/warehouse/test_employees/*
1Pierre50000.0
2Marie60000.0
3Jean55000.0
yessine@worker1:~$
yessine@worker2: ~
Fichier Actions Éditer Vue Aide
yessine@worker2: ~
yessine@worker2:~$ jps
2819 RunJar
3796 Jps
2282 NodeManager
2143 DataNode
yessine@worker2:~$

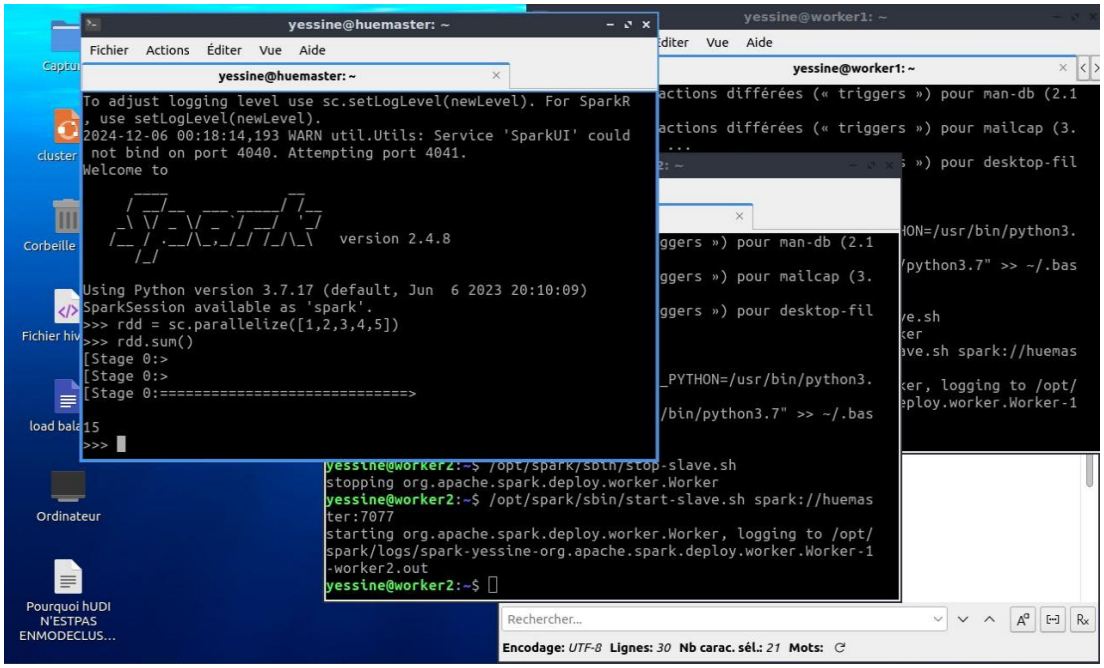
Test hive



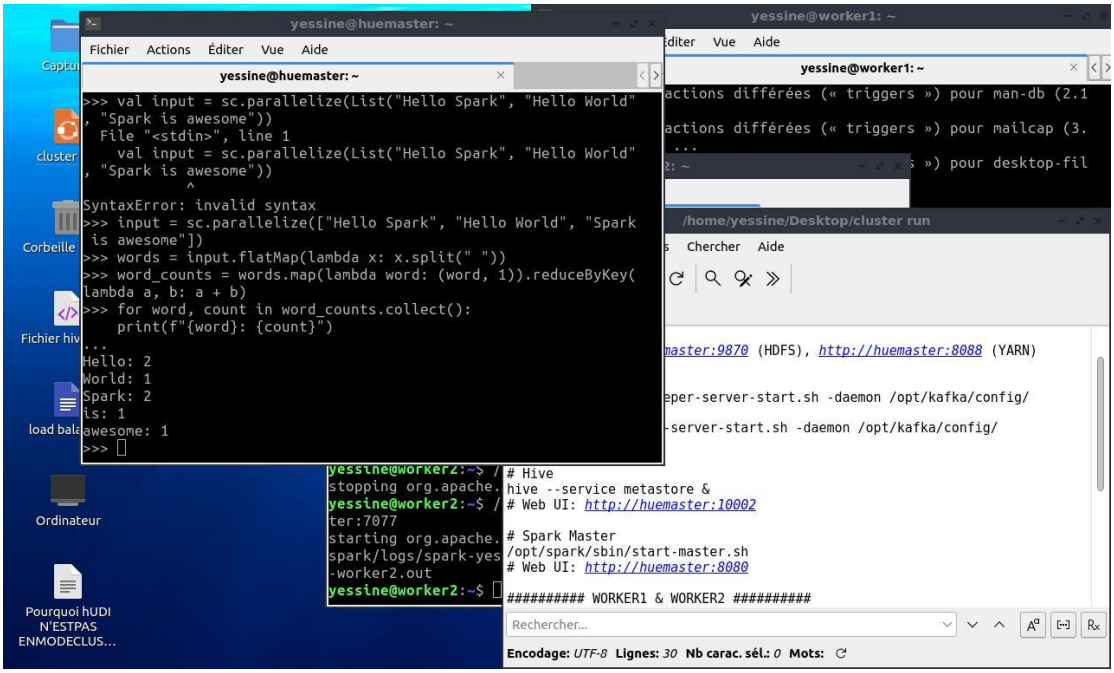
Spark

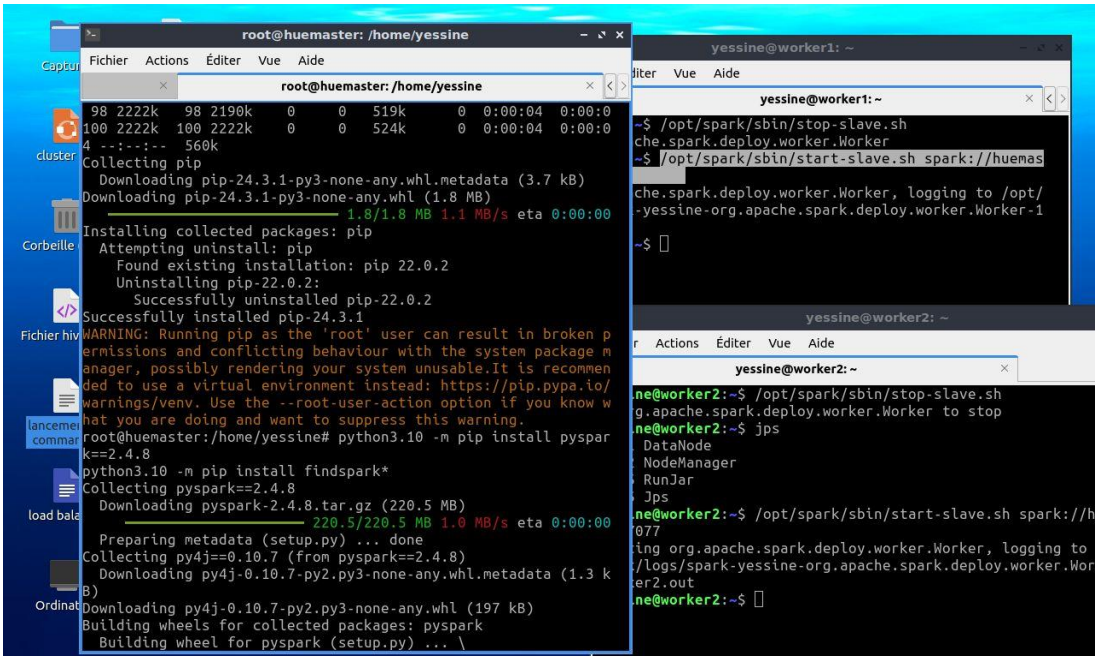



yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yessine@huemaster: ~
export SPARK_WORKER_C
export SPARK_WORKER_M
EOF
yessine@huemaster:~$
< EOF
spark.master
spark.eventLog.enable
spark.eventLog.dir
spark.serializer
erializer
spark.driver.memory
spark.executor.memory
EOF
yessine@huemaster:~$
starting org.apache.s
spark/logs/spark-yess
-huemaster.out
yessine@huemaster:~$
11942 Jps
11852 Master
yessine@huemaster:~$
yessine@worker2: ~
Fichier Actions Éditer Vue Aide
yessine@worker2: ~
ice.sh
-rwxrwxrwx 1 501 yessine 3151 2021 8 ماي
-rwxrwxrwx 1 501 yessine 1527 2021 8 ماي
-rwxrwxrwx 1 501 yessine 1857 2021 8 ماي
.sh
-rwxrwxrwx 1 501 yessine 1478 2021 8 ماي
-rwxrwxrwx 1 501 yessine 1056 2021 8 ماي
r.sh
-rwxrwxrwx 1 501 yessine 1080 2021 8 ماي
-rwxrwxrwx 1 501 yessine 1227 2021 8 ماي
her.sh
-rwxrwxrwx 1 501 yessine 1084 2021 8 ماي
-service.sh
-rwxrwxrwx 1 501 yessine 1067 2021 8 ماي stop-shuffle-servi
ce.sh
-rwxrwxrwx 1 501 yessine 1557 2021 8 ماي stop-slave.sh
-rwxrwxrwx 1 501 yessine 1064 2021 8 ماي stop-slaves.sh
-rwxrwxrwx 1 501 yessine 1066 2021 8 ماي stop-thriftserver.
sh
yessine@worker2:~$ /opt/spark/sbin/start-slave.sh spark://huemas
ter:7077
starting org.apache.spark.deploy.worker.Worker, logging to /opt/
spark/logs/spark-yessine-org.apache.spark.deploy.worker.Worker-1
-worker2.out
yessine@worker2:~$ jps
6311 Worker
6363 Jps
yessine@worker2:~$
yessine@worker1: ~
Fichier Actions Éditer Vue Aide
yessine@worker1: ~
spark.driver.memory 2g
spark.executor.memory 2g
EOF
yessine@worker1:~$ /opt/spark/sbin/start-worker.sh spark://hu
ster:7077
bash: /opt/spark/sbin/start-worker.sh: Aucun fichier ou dossi
de ce nom
yessine@worker1:~$ /opt/spark/sbin/start-slave.sh spark://hue
ter:7077
starting org.apache.spark.deploy.worker.Worker, logging to /o
spark/logs/spark-yessine-org.apache.spark.deploy.worker.Worke
-worker1.out
yessine@worker1:~$ jps
5396 Worker
5448 Jps
yessine@worker1:~$
Pourquoi hUDI N'ESTPAS ENMODECLUS...
Réseau


yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yessine@huemaster: ~
To adjust logging level use sc.setLogLevel(newLevel). For SparkR
, use setLogLevel(newLevel).
2024-12-06 00:18:14,193 WARN util.Utils: Service 'SparkUI' could
not bind on port 4040. Attempting port 4041.
Welcome to
version 2.4.8
Using Python version 3.7.17 (default, Jun 6 2023 20:10:09)
SparkSession available as 'spark'.
>>> rdd = sc.parallelize([1,2,3,4,5])
>>> rdd.sum()
[Stage 0:>
[Stage 0:>
[Stage 0:==============================>
15
>>>
yessine@worker1: ~
Fichier Actions Éditer Vue Aide
yessine@worker1: ~
actions différées (« triggers ») pour man-db (2.1
actions différées (« triggers ») pour mailcap (3.
actions différées (« triggers ») pour desktop-fil
yessine@worker2:~$ /opt/spark/sbin/stop-slave.sh
stopping org.apache.spark.deploy.worker.Worker
yessine@worker2:~$ /opt/spark/sbin/start-slave.sh spark://huemas
ter:7077
starting org.apache.spark.deploy.worker.Worker, logging to /opt/
spark/logs/spark-yessine-org.apache.spark.deploy.worker.Worker-1
-worker2.out
yessine@worker2:~$
Rechercher...
Encodage: UTF-8 Lignes: 30 Nb carac. sél.: 21 Mots:
Ordinateur
Pourquoi hUDI N'ESTPAS ENMODECLUS...

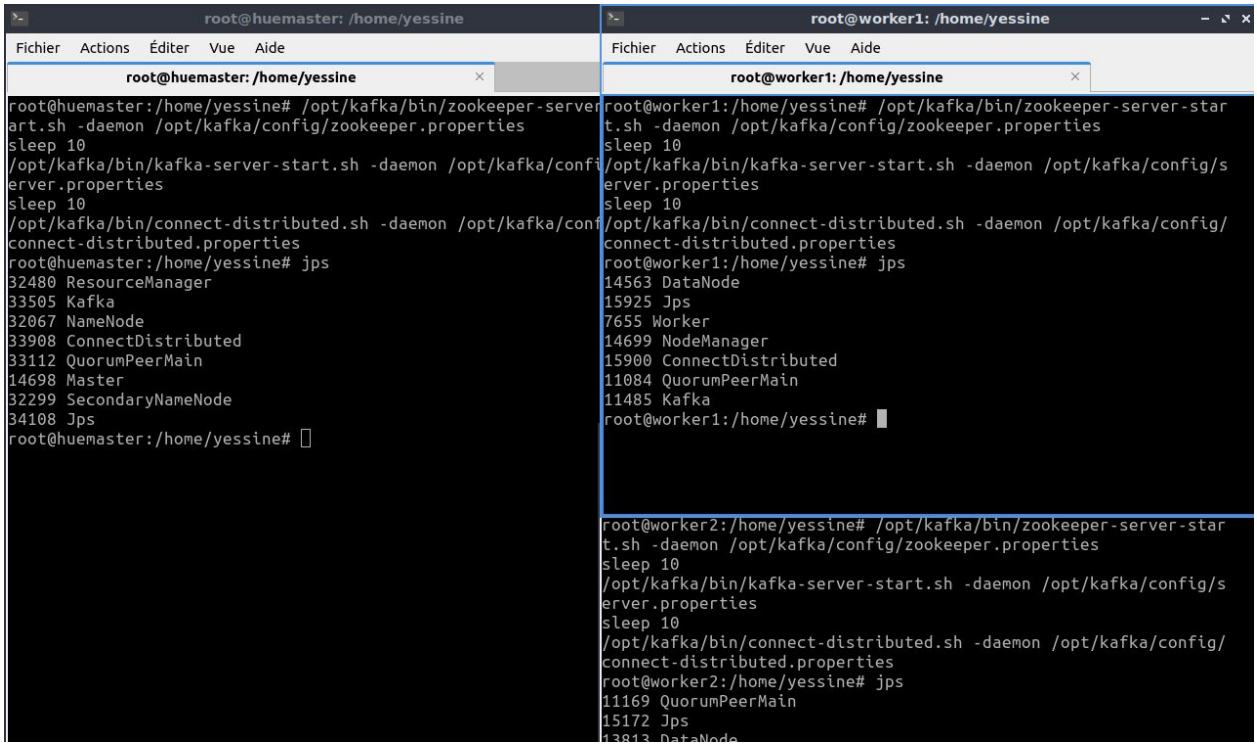
Spark Test

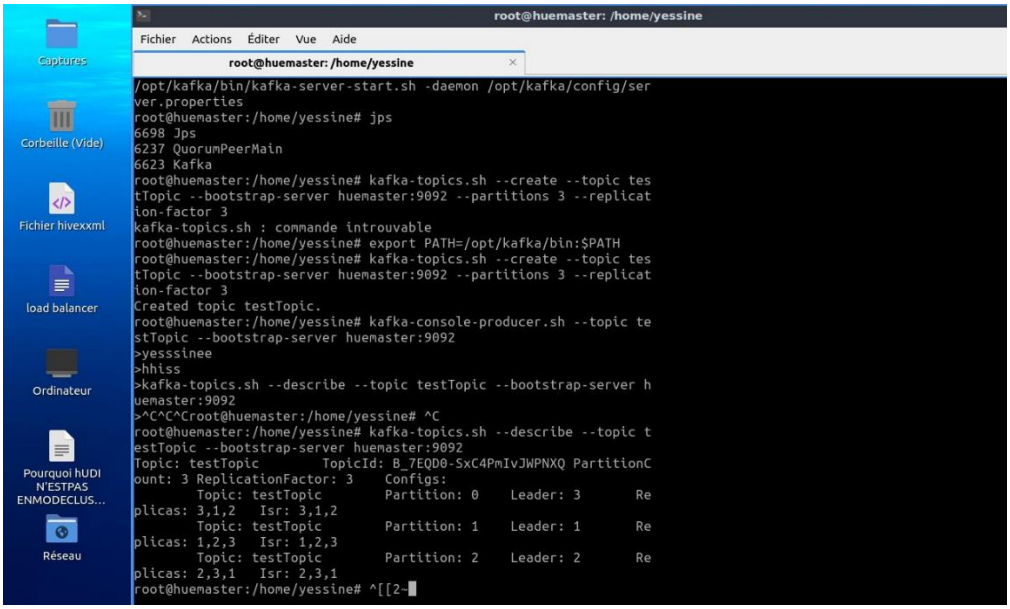

yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yessine@huemaster: ~
>>> val input = sc.parallelize(List("Hello Spark", "Hello World", "Spark is awesome"))
File "<stdin>", line 1
val input = sc.parallelize(List("Hello Spark", "Hello World", "Spark is awesome"))
^
SyntaxError: invalid syntax
>>> input = sc.parallelize(["Hello Spark", "Hello World", "Spark is awesome"])
>>> words = input.flatMap(lambda x: x.split(" "))
>>> word_counts = words.map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
>>> for word, count in word_counts.collect():
print(f"{word}: {count}")
...
Hello: 2
World: 1
Spark: 2
is: 1
awesome: 1
>>>
yessine@worker1: ~
yessine@worker1: ~
/home/yessine/Desktop/cluster run
Chercher Aide
yessine@worker2:~$ /
stopping org.apache.
yessine@worker2:~$ /
ter:7077
starting org.apache.
spark/logs/spark-yes
-worker2.out
yessine@worker2:~$
Hive
hive --service metastore &
Web UI: http://huemaster:10002
Spark Master
/opt/spark/sbin/start-master.sh
Web UI: http://huemaster:8080
########## WORKER1 & WORKER2 ##########
Encodage: UTF-8 Lignes: 30 Nb carac. sél.: 0 Mots:


root@huemaster: /home/yessine
Fichier Actions Éditer Vue Aide
root@huemaster: /home/yessine
Collecting pip
Downloading pip-24.3.1-py3-none-any.whl.metadata (3.7 kB)
Downloading pip-24.3.1-py3-none-any.whl (1.8 MB)
1.8/1.8 MB 1.1 MB/s eta 0:00:00
Installing collected packages: pip
Attempting uninstall: pip
Found existing installation: pip 22.0.2
Uninstalling pip-22.0.2:
Successfully uninstalled pip-22.0.2
Successfully installed pip-24.3.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system unusable.It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning.
root@huemaster:/home/yessine# python3.10 -m pip install pyspark==2.4.8
python3.10 -m pip install findspark*
Collecting pyspark==2.4.8
Downloading pyspark-2.4.8.tar.gz (220.5 MB)
220.5/220.5 MB 1.0 MB/s eta 0:00:00
Preparing metadata (setup.py) ... done
Collecting py4j==0.10.7 (from pyspark==2.4.8)
Downloading py4j-0.10.7-py2.py3-none-any.whl.metadata (1.3 kB)
Downloading py4j-0.10.7-py2.py3-none-any.whl (197 kB)
Building wheels for collected packages: pyspark
Building wheel for pyspark (setup.py) ... \
yessine@worker1: ~
yessine@worker1: ~
~$ /opt/spark/sbin/stop-slave.sh
~$ /opt/spark/sbin/start-slave.sh spark://huemas
yessine@worker2: ~
yessine@worker2: ~
~$ /opt/spark/sbin/stop-slave.sh
g.apache.spark.deploy.worker.Worker to stop
~$ jps
DataNode
NodeManager
RunJar
Jps
~$ /opt/spark/sbin/start-slave.sh spark://h
~$

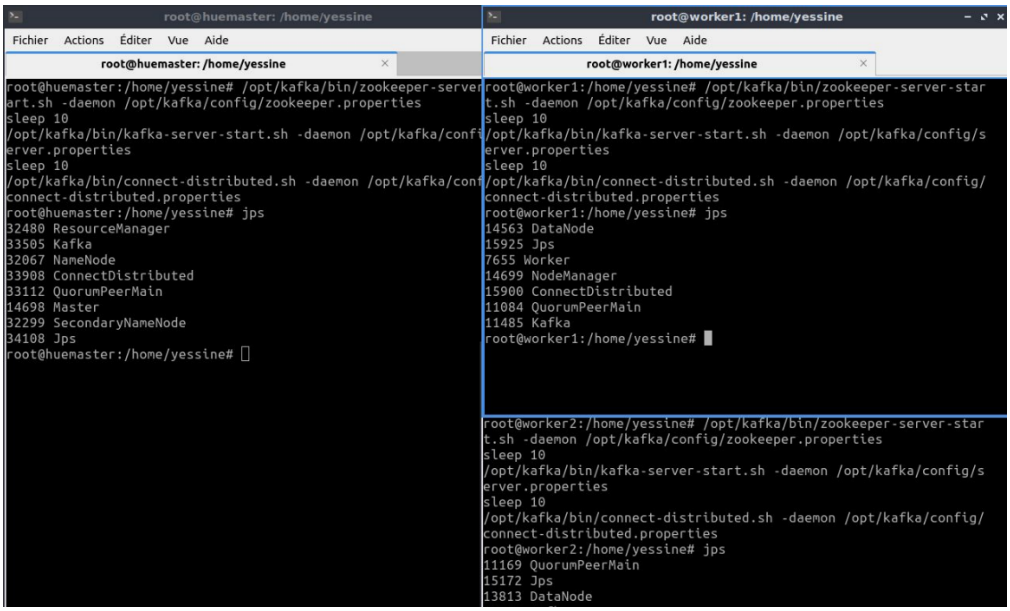

Spark shell - Spark Jobs — Mozilla Firefox
huemaster:8088/proxy/application_1733609494086_0001/
Spark 2.4.8
Jobs
Stages
Storage
Environment
Executors
Spark shell application UI
Spark Jobs (?)
User: yessine
Total Uptime: 6,6 min
Scheduling Mode: FIFO
Completed Jobs: 1
Event Timeline
Completed Jobs (1)
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
0
count at <console>:25
count at <console>:25
2024/12/07 23:31:51
6 s
1/1
2/2

Kafka

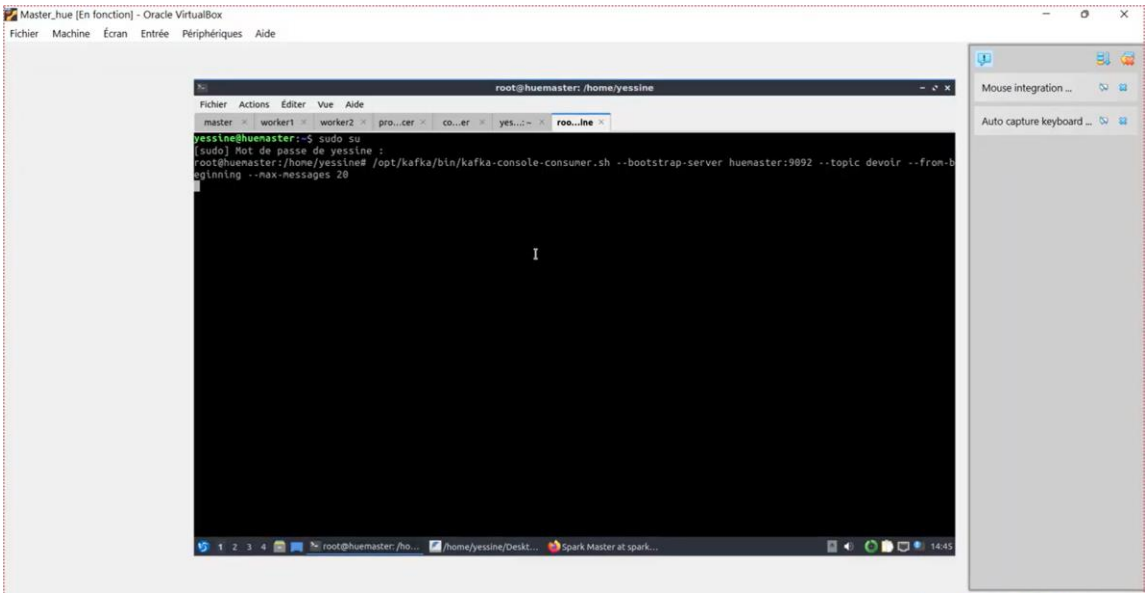

root@huemaster: /home/yessine
Fichier Actions Éditer Vue Aide
root@huemaster: /home/yessine
root@huemaster:/home/yessine# /opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties
sleep 10
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
sleep 10
/opt/kafka/bin/connect-distributed.sh -daemon /opt/kafka/config/connect-distributed.properties
root@huemaster:/home/yessine# jps
32480 ResourceManager
33505 Kafka
32067 NameNode
33908 ConnectDistributed
33112 QuorumPeerMain
14698 Master
32299 SecondaryNameNode
34108 Jps
root@huemaster:/home/yessine#
root@worker1: /home/yessine
Fichier Actions Éditer Vue Aide
root@worker1: /home/yessine
root@worker1:/home/yessine# /opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties
sleep 10
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
sleep 10
/opt/kafka/bin/connect-distributed.sh -daemon /opt/kafka/config/connect-distributed.properties
root@worker1:/home/yessine# jps
14563 DataNode
15925 Jps
7655 Worker
14699 NodeManager
15900 ConnectDistributed
11084 QuorumPeerMain
11485 Kafka
root@worker1:/home/yessine#
root@worker2:/home/yessine# /opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties
sleep 10
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
sleep 10
/opt/kafka/bin/connect-distributed.sh -daemon /opt/kafka/config/connect-distributed.properties
root@worker2:/home/yessine# jps
11169 QuorumPeerMain
15172 Jps

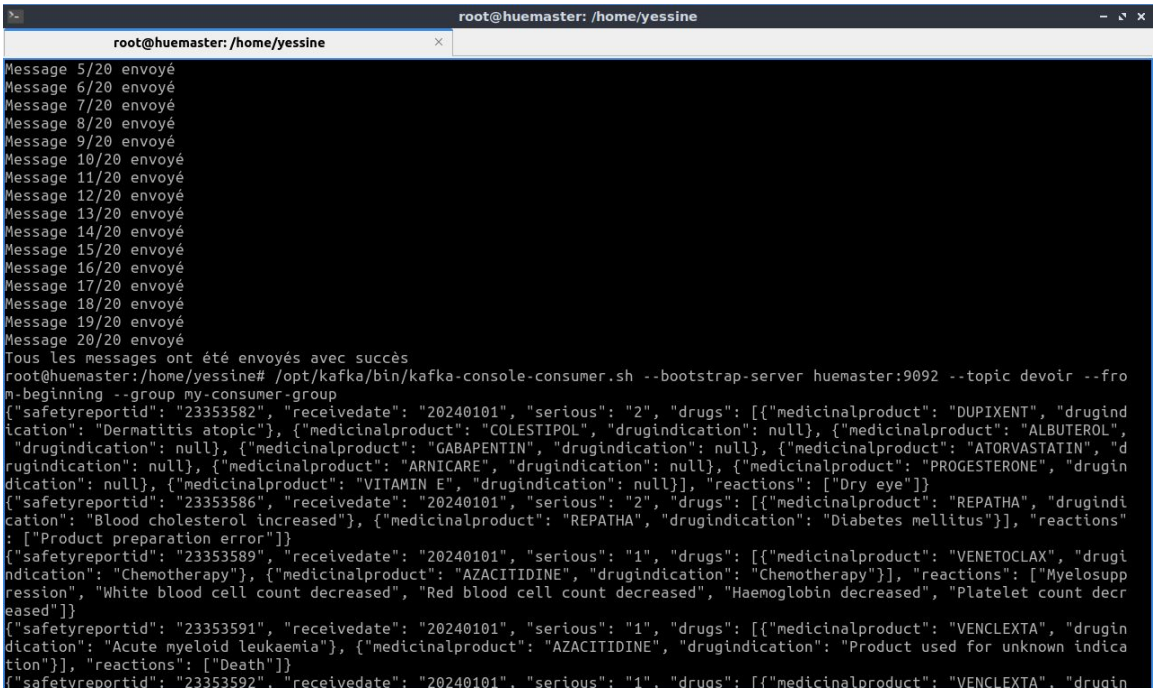

root@huemaster: /home/yessine
Fichier Actions Éditer Vue Aide
root@huemaster: /home/yessine
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
root@huemaster:/home/yessine# jps
6698 Jps
6237 QuorumPeerMain
6623 Kafka
root@huemaster:/home/yessine# kafka-topics.sh --create --topic testTopic --bootstrap-server huemaster:9092 --partitions 3 --replication-factor 3
kafka-topics.sh : commande introuvable
root@huemaster:/home/yessine# export PATH=/opt/kafka/bin:$PATH
root@huemaster:/home/yessine# kafka-topics.sh --create --topic testTopic --bootstrap-server huemaster:9092 --partitions 3 --replication-factor 3
Created topic testTopic.
root@huemaster:/home/yessine# kafka-console-producer.sh --topic testTopic --bootstrap-server huemaster:9092
>yesssinee
>hhiss
>kafka-topics.sh --describe --topic testTopic --bootstrap-server huemaster:9092
>^C^C^Croot@huemaster:/home/yessine# ^C
root@huemaster:/home/yessine# kafka-topics.sh --describe --topic testTopic --bootstrap-server huemaster:9092
Topic: testTopic TopicId: B_7EQD0-SxC4PmIvJWPNXQ PartitionCount: 3 ReplicationFactor: 3 Configs:
Topic: testTopic Partition: 0 Leader: 3 Replicas: 3,1,2 Isr: 3,1,2
Topic: testTopic Partition: 1 Leader: 1 Replicas: 1,2,3 Isr: 1,2,3
Topic: testTopic Partition: 2 Leader: 2 Replicas: 2,3,1 Isr: 2,3,1
root@huemaster:/home/yessine# ^[[2-
Captures
Corbeille (Vide)
Fichier hivexxml
load balancer
Ordinateur
Pourquoi hUDI N'ESTPAS ENMODECLUS...
Réseau

JPS


root@huemaster: /home/yessine
Fichier Actions Éditer Vue Aide
root@huemaster: /home/yessine
root@huemaster:/home/yessine# /opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties
sleep 10
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
sleep 10
/opt/kafka/bin/connect-distributed.sh -daemon /opt/kafka/config/connect-distributed.properties
root@huemaster:/home/yessine# jps
32480 ResourceManager
33505 Kafka
32067 NameNode
33908 ConnectDistributed
33112 QuorumPeerMain
14698 Master
32299 SecondaryNameNode
34108 Jps
root@huemaster:/home/yessine#
root@worker1: /home/yessine
Fichier Actions Éditer Vue Aide
root@worker1: /home/yessine
root@worker1:/home/yessine# /opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties
sleep 10
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
sleep 10
/opt/kafka/bin/connect-distributed.sh -daemon /opt/kafka/config/connect-distributed.properties
root@worker1:/home/yessine# jps
14563 DataNode
15925 Jps
7655 Worker
14699 NodeManager
15900 ConnectDistributed
11084 QuorumPeerMain
11485 Kafka
root@worker1:/home/yessine#
root@worker2:/home/yessine# /opt/kafka/bin/zookeeper-server-start.sh -daemon /opt/kafka/config/zookeeper.properties
sleep 10
/opt/kafka/bin/kafka-server-start.sh -daemon /opt/kafka/config/server.properties
sleep 10
/opt/kafka/bin/connect-distributed.sh -daemon /opt/kafka/config/connect-distributed.properties
root@worker2:/home/yessine# jps
11169 QuorumPeerMain
15172 Jps
13813 DataNode

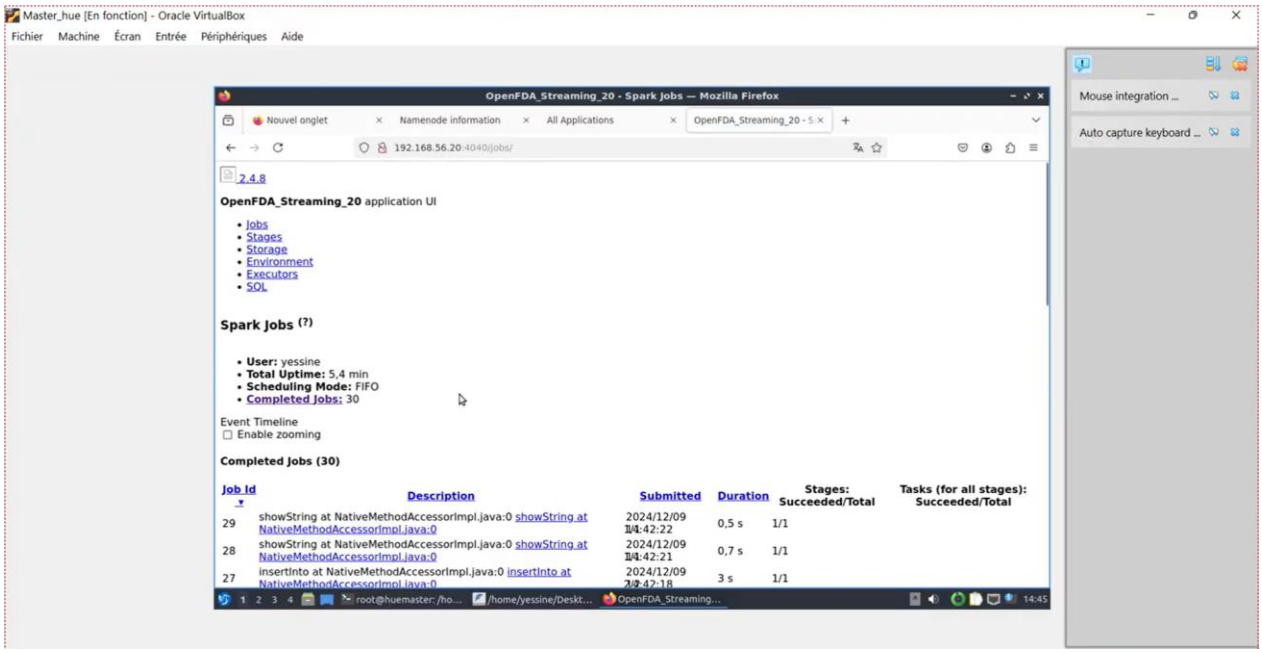
producer

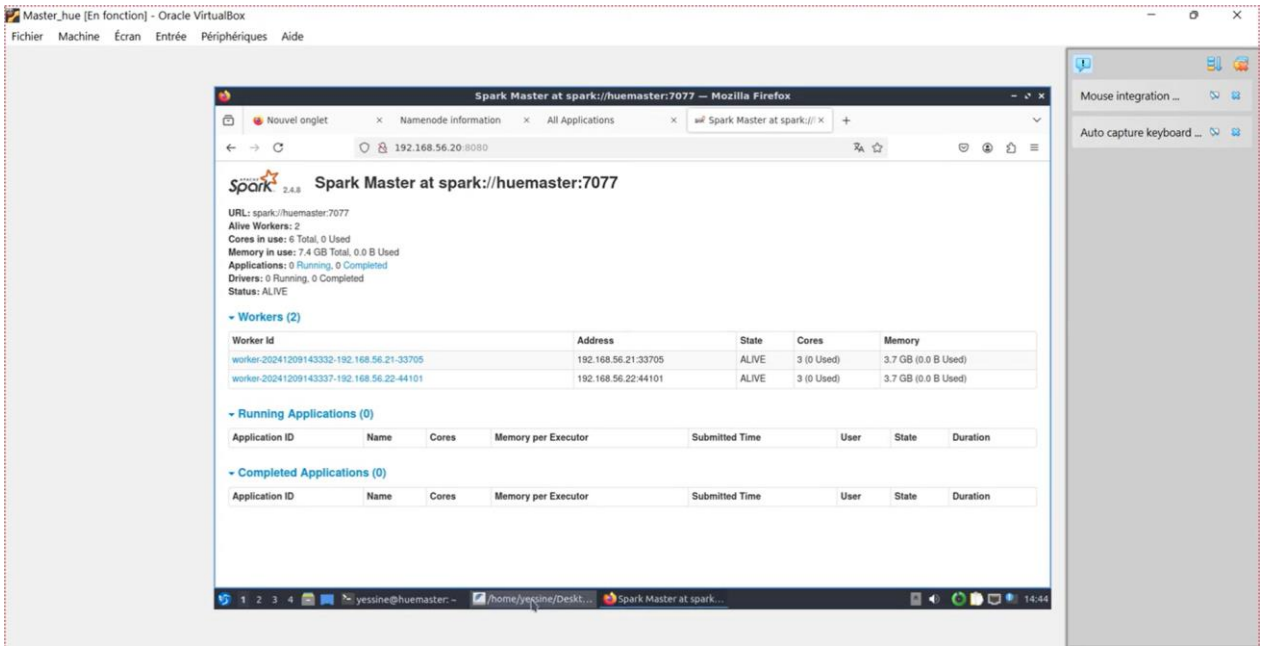

Master_hue [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
root@huemaster: /home/yessine
Fichier Actions Éditer Vue Aide
master worker1 worker2 pro...cer co...er yes...:~ roo...ine
yessine@huemaster:~$ sudo su
[sudo] Mot de passe de yessine :
root@huemaster:/home/yessine# /opt/kafka/bin/kafka-console-consumer.sh --bootstrap-server huemaster:9092 --topic devoir --from-b
eginning --max-messages 20
Mouse integration ...
Auto capture keyboard ...

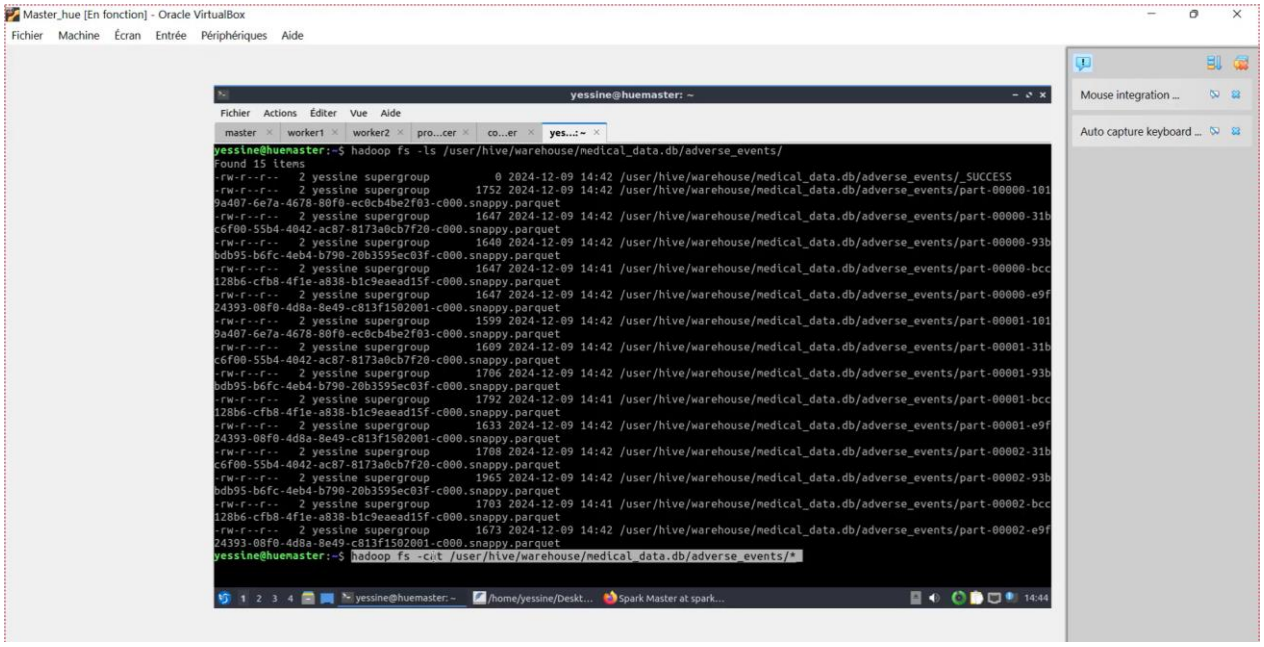

root@huemaster: /home/yessine
root@huemaster: /home/yessine
Message 5/20 envoyé
Message 6/20 envoyé
Message 7/20 envoyé
Message 8/20 envoyé
Message 9/20 envoyé
Message 10/20 envoyé
Message 11/20 envoyé
Message 12/20 envoyé
Message 13/20 envoyé
Message 14/20 envoyé
Message 15/20 envoyé
Message 16/20 envoyé
Message 17/20 envoyé
Message 18/20 envoyé
Message 19/20 envoyé
Message 20/20 envoyé
Tous les messages ont été envoyés avec succès
root@huemaster:/home/yessine# /opt/kafka/bin/kafka-console-consumer.sh --bootstrap-server huemaster:9092 --topic devoir --fro
m-beginning --group my-consumer-group
{"safetyreportid": "23353582", "receivedate": "20240101", "serious": "2", "drugs": [{"medicinalproduct": "DUPIXENT", "drugind
ication": "Dermatitis atopic"}, {"medicinalproduct": "COLESTIPOL", "drugindication": null}, {"medicinalproduct": "ALBUTEROL",
"drugindication": null}, {"medicinalproduct": "GABAPENTIN", "drugindication": null}, {"medicinalproduct": "ATORVASTATIN", "d
rugindication": null}, {"medicinalproduct": "ARNICARE", "drugindication": null}, {"medicinalproduct": "PROGESTERONE", "drugin
dication": null}, {"medicinalproduct": "VITAMIN E", "drugindication": null}], "reactions": ["Dry eye"]}
{"safetyreportid": "23353586", "receivedate": "20240101", "serious": "2", "drugs": [{"medicinalproduct": "REPATHA", "drugindi
cation": "Blood cholesterol increased"}, {"medicinalproduct": "REPATHA", "drugindication": "Diabetes mellitus"}], "reactions"
: ["Product preparation error"]}
{"safetyreportid": "23353589", "receivedate": "20240101", "serious": "1", "drugs": [{"medicinalproduct": "VENETOCLAX", "drugi
ndication": "Chemotherapy"}, {"medicinalproduct": "AZACITIDINE", "drugindication": "Chemotherapy"}], "reactions": ["Myelosupp
ression", "White blood cell count decreased", "Red blood cell count decreased", "Haemoglobin decreased", "Platelet count decr
eased"]}
{"safetyreportid": "23353591", "receivedate": "20240101", "serious": "1", "drugs": [{"medicinalproduct": "VENCLEXTA", "drugin
dication": "Acute myeloid leukaemia"}, {"medicinalproduct": "AZACITIDINE", "drugindication": "Product used for unknown indica
tion"}], "reactions": ["Death"]}

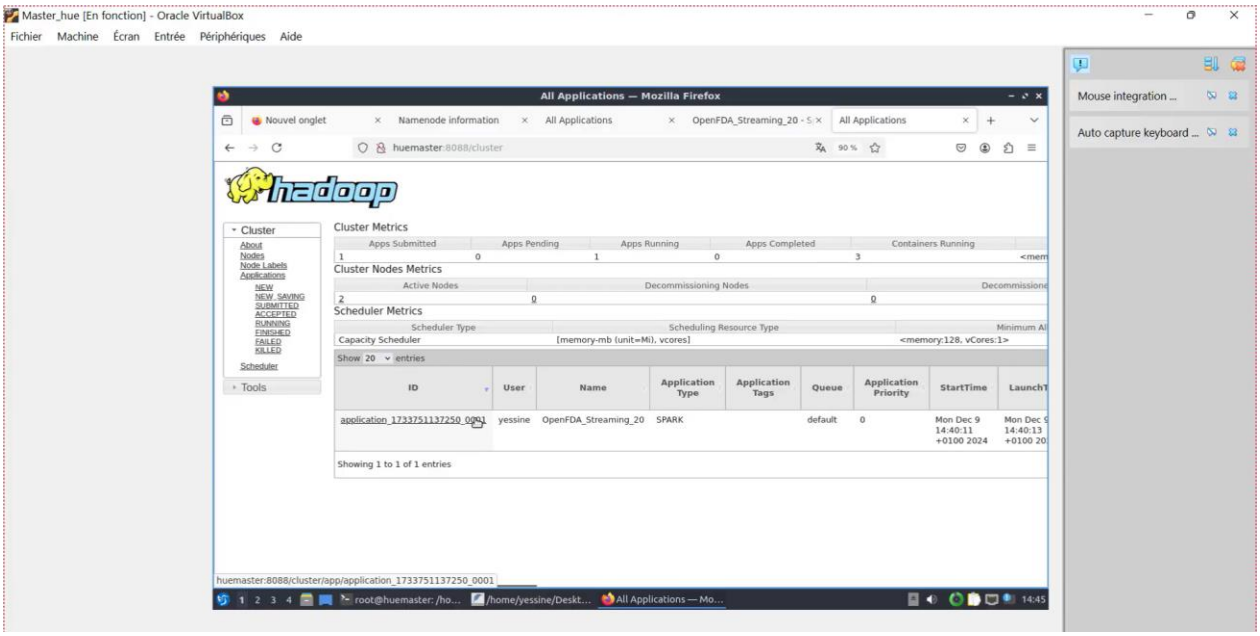
consumer

```
2024-12-08 23:59:07,126 WARN streaming.ProcessingTimeExecutor: Current batch is falling behind. The trigger interval is 5000
milliseconds, but spent 27126 milliseconds

Traitement batch 135 - 4 messages

Données reçues :
+--------------+-----------+-------+---------------+---------------------------------+
|safetyreportid|receivedate|serious|medicinalproduct|drugindication                  |
+--------------+-----------+-------+---------------+---------------------------------+
|23353590      |20240101   |true   |VENCLEXTA      |Acute myeloid leukaemia          |
|23353590      |20240101   |true   |AZACITIDINE    |Product used for unknown indication|
|23353586      |20240101   |false  |REPATHA        |Blood cholesterol increased      |
|23353586      |20240101   |false  |REPATHA        |Diabetes mellitus                |
|23353589      |20240101   |true   |VENETOCLAX     |Chemotherapy                     |
+--------------+-----------+-------+---------------+---------------------------------+
only showing top 5 rows
```



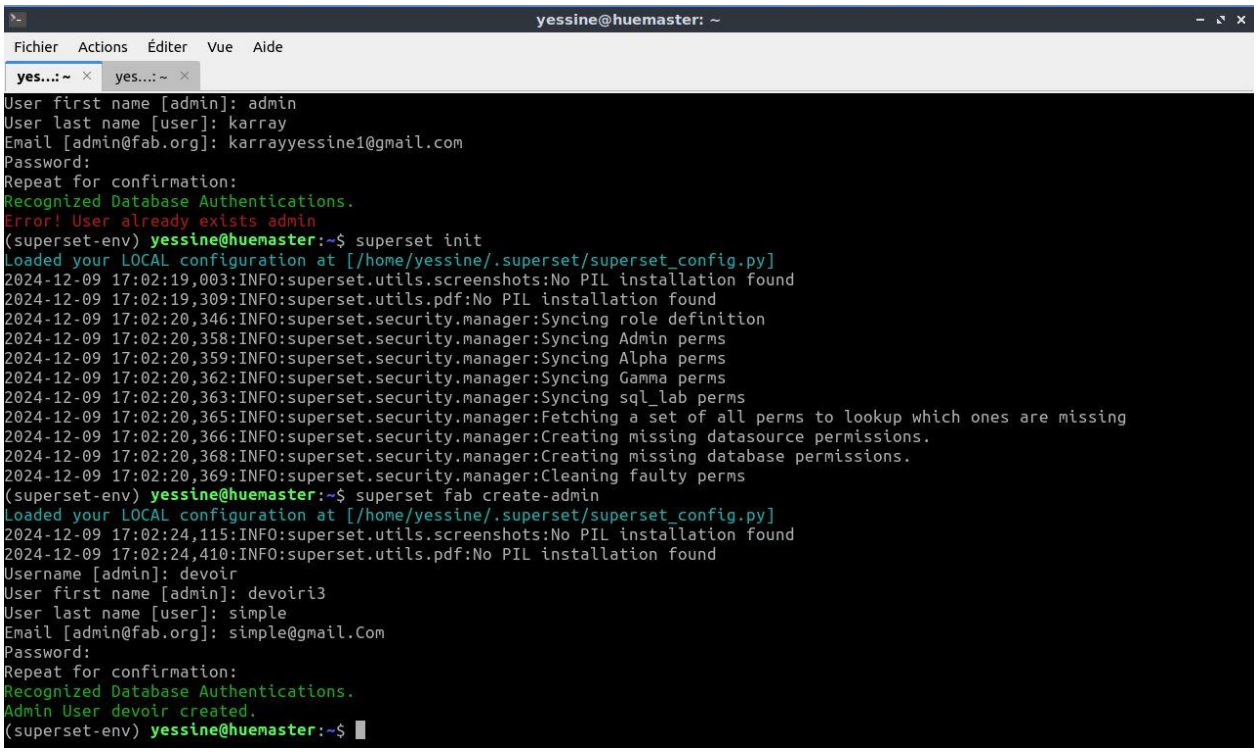

Master_hue [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Spark Master at spark://huemaster:7077 — Mozilla Firefox
Nouvel onglet
Namenode information
All Applications
Spark Master at spark://
192.168.56.20:8080
Spark Master at spark://huemaster:7077
URL: spark://huemaster:7077
Alive Workers: 2
Cores in use: 6 Total, 0 Used
Memory in use: 7.4 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE
Workers (2)
Worker Id
Address
State
Cores
Memory
worker-20241209143332-192.168.56.21-33705
192.168.56.21:33705
ALIVE
3 (0 Used)
3.7 GB (0.0 B Used)
worker-20241209143337-192.168.56.22-44101
192.168.56.22:44101
ALIVE
3 (0 Used)
3.7 GB (0.0 B Used)
Running Applications (0)
Application ID
Name
Cores
Memory per Executor
Submitted Time
User
State
Duration
Completed Applications (0)
Application ID
Name
Cores
Memory per Executor
Submitted Time
User
State
Duration
yessine@huemaster: ~
/home/yessine/Deskt...
Spark Master at spark...
14:44
Mouse integration ...
Auto capture keyboard ...

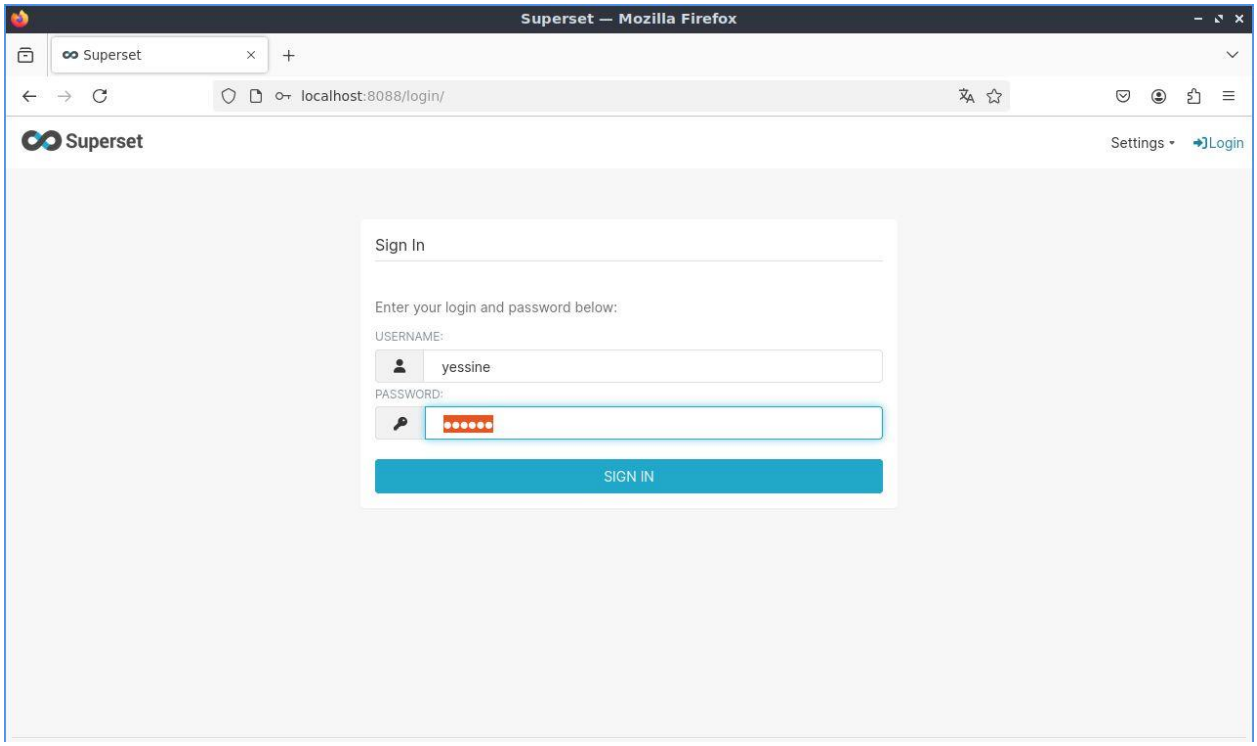

Master_hue [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
master
worker1
worker2
pro...cer
co...er
yes...: ~
yessine@huenaster:~$ hadoop fs -ls /user/hive/warehouse/medical_data.db/adverse_events/
Found 15 items
-rw-r--r-- 2 yessine supergroup 0 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/_SUCCESS
-rw-r--r-- 2 yessine supergroup 1752 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00000-101
9a407-6e7a-4678-80f0-ec0cb4be2f03-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1647 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00000-31b
c6f00-55b4-4042-ac87-8173a0cb7f20-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1640 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00000-93b
bdb95-b6fc-4eb4-b790-20b3595ec03f-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1647 2024-12-09 14:41 /user/hive/warehouse/medical_data.db/adverse_events/part-00000-bcc
128b6-cfb8-4f1e-a838-b1c9eaead15f-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1647 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00000-e9f
24393-08f0-4d8a-8e49-c813f1502001-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1599 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00001-101
9a407-6e7a-4678-80f0-ec0cb4be2f03-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1609 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00001-31b
c6f00-55b4-4042-ac87-8173a0cb7f20-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1706 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00001-93b
bdb95-b6fc-4eb4-b790-20b3595ec03f-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1792 2024-12-09 14:41 /user/hive/warehouse/medical_data.db/adverse_events/part-00001-bcc
128b6-cfb8-4f1e-a838-b1c9eaead15f-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1633 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00001-e9f
24393-08f0-4d8a-8e49-c813f1502001-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1708 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00002-31b
c6f00-55b4-4042-ac87-8173a0cb7f20-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1965 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00002-93b
bdb95-b6fc-4eb4-b790-20b3595ec03f-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1703 2024-12-09 14:41 /user/hive/warehouse/medical_data.db/adverse_events/part-00002-bcc
128b6-cfb8-4f1e-a838-b1c9eaead15f-c000.snappy.parquet
-rw-r--r-- 2 yessine supergroup 1673 2024-12-09 14:42 /user/hive/warehouse/medical_data.db/adverse_events/part-00002-e9f
24393-08f0-4d8a-8e49-c813f1502001-c000.snappy.parquet
yessine@huenaster:~$ hadoop fs -cat /user/hive/warehouse/medical_data.db/adverse_events/*
yessine@huemaster:~
/home/yessine/Deskt...
Spark Master at spark...
14:44
Mouse integration ...
Auto capture keyboard ...

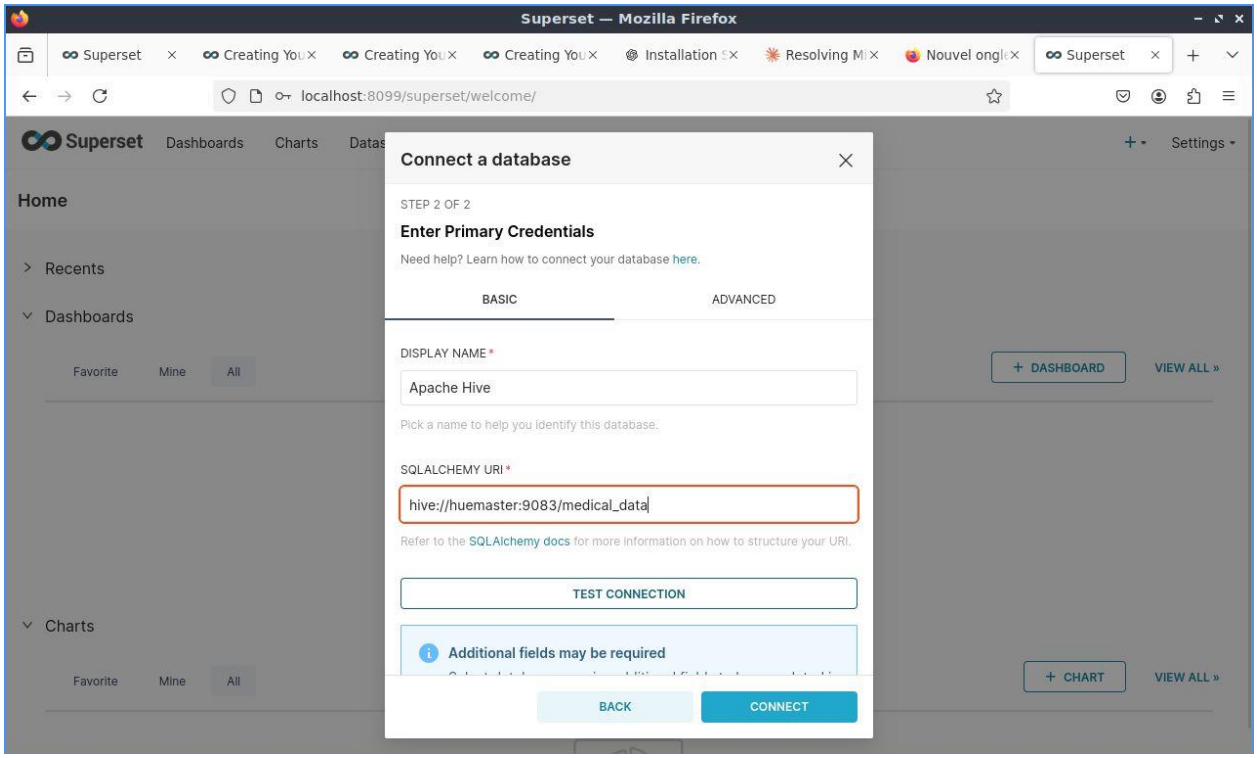


Superset

```
yessine@huemaster: ~
Fichier  Actions  Éditer  Vue  Aide
yessine@huemaster:~$ python3 -m venv superset-env
source superset-env/bin/activate
(superset-env) yessine@huemaster:~$ pip install --upgrade pip setuptools
Requirement already satisfied: pip in ./superset-env/lib/python3.10/site-packages (22.0.2)
Collecting pip
  Downloading pip-24.3.1-py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 1.1 MB/s eta 0:00:00
Requirement already satisfied: setuptools in ./superset-env/lib/python3.10/site-packages (59.6.0)
Collecting setuptools
  Downloading setuptools-75.6.0-py3-none-any.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 1.1 MB/s eta 0:00:00
Installing collected packages: setuptools, pip
  Attempting uninstall: setuptools
    Found existing installation: setuptools 59.6.0
    Uninstalling setuptools-59.6.0:
      Successfully uninstalled setuptools-59.6.0
  Attempting uninstall: pip
    Found existing installation: pip 22.0.2
    Uninstalling pip-22.0.2:
      Successfully uninstalled pip-22.0.2
Successfully installed pip-24.3.1 setuptools-75.6.0
(superset-env) yessine@huemaster:~$ pip install apache-superset
Collecting apache-superset
  Downloading apache-superset-4.1.1.tar.gz (55.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.8/55.5 MB 1.1 MB/s eta 0:00:45
```


yessine@huemaster: ~
Fichier Actions Éditer Vue Aide
yes...: ~
yes...: ~
User first name [admin]: admin
User last name [user]: karray
Email [admin@fab.org]: karrayyessine1@gmail.com
Password:
Repeat for confirmation:
Recognized Database Authentications.
Error! User already exists admin
(superset-env) yessine@huemaster:~$ superset init
Loaded your LOCAL configuration at [/home/yessine/.superset/superset_config.py]
2024-12-09 17:02:19,003:INFO:superset.utils.screenshots:No PIL installation found
2024-12-09 17:02:19,309:INFO:superset.utils.pdf:No PIL installation found
2024-12-09 17:02:20,346:INFO:superset.security.manager:Syncing role definition
2024-12-09 17:02:20,358:INFO:superset.security.manager:Syncing Admin perms
2024-12-09 17:02:20,359:INFO:superset.security.manager:Syncing Alpha perms
2024-12-09 17:02:20,362:INFO:superset.security.manager:Syncing Gamma perms
2024-12-09 17:02:20,363:INFO:superset.security.manager:Syncing sql_lab perms
2024-12-09 17:02:20,365:INFO:superset.security.manager:Fetching a set of all perms to lookup which ones are missing
2024-12-09 17:02:20,366:INFO:superset.security.manager:Creating missing datasource permissions.
2024-12-09 17:02:20,368:INFO:superset.security.manager:Creating missing database permissions.
2024-12-09 17:02:20,369:INFO:superset.security.manager:Cleaning faulty perms
(superset-env) yessine@huemaster:~$ superset fab create-admin
Loaded your LOCAL configuration at [/home/yessine/.superset/superset_config.py]
2024-12-09 17:02:24,115:INFO:superset.utils.screenshots:No PIL installation found
2024-12-09 17:02:24,410:INFO:superset.utils.pdf:No PIL installation found
Username [admin]: devoir
User first name [admin]: devoiri3
User last name [user]: simple
Email [admin@fab.org]: simple@gmail.Com
Password:
Repeat for confirmation:
Recognized Database Authentications.
Admin User devoir created.
(superset-env) yessine@huemaster:~$


Superset — Mozilla Firefox
Superset
localhost:8088/login/
Superset
Settings
Login
Sign In
Enter your login and password below:
USERNAME:
yessine
PASSWORD:
SIGN IN


Superset — Mozilla Firefox
localhost:8099/superset/welcome/
Superset
Dashboards
Charts
Settings
Home
Recents
Dashboards
Favorite
Mine
All
+ DASHBOARD
VIEW ALL »
Charts
+ CHART
Connect a database
STEP 2 OF 2
Enter Primary Credentials
Need help? Learn how to connect your database here.
BASIC
ADVANCED
DISPLAY NAME *
Apache Hive
Pick a name to help you identify this database.
SQLALCHEMY URI *
hive://huemaster:9083/medical_data
Refer to the SQLAlchemy docs for more information on how to structure your URI.
TEST CONNECTION
Additional fields may be required
BACK
CONNECT

Architecture Apache Doris pour l'Analyse des Données OpenFDA

6. Architecture du Système

6.1 Vue d'Ensemble

Le système repose sur une architecture distribuée comprenant :

- 2 machines physiques
- Facteur de réplication de 2 backend
- Architecture en trois couches (Frontend, Backend, Storage)

6.2 Composants Principaux

6.2.1 Frontend (FE)

- Gestion des requêtes SQL
- Optimisation des plans d'exécution
- Gestion des métadonnées
- Coordination des backends

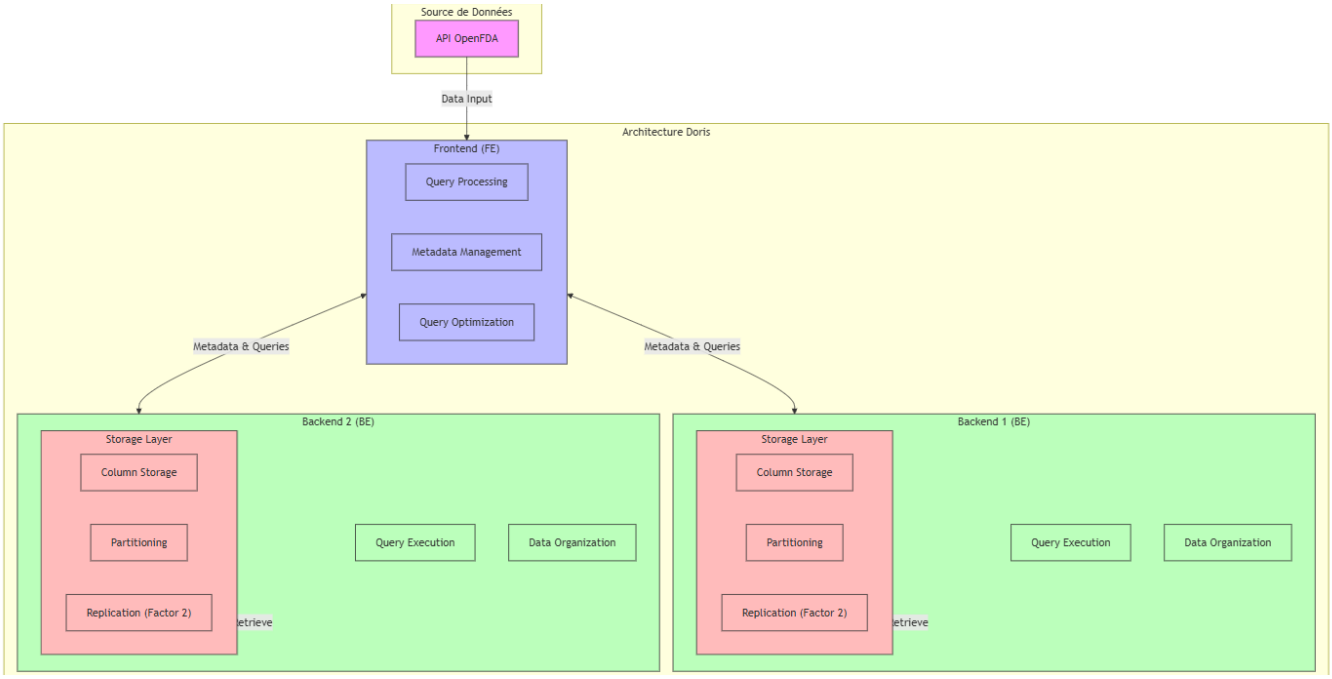
6.2.2 Backend (BE)

- Exécution des requêtes
- Stockage physique des données
- Gestion de la compression
- Indexation

6.2.3 Storage Layer

- Stockage columnaire
- Partitionnement des données
- Gestion des segments
- Réplication

Notre deuxième Architecture



- Doris est un système de base de données analytique distribué conçu pour offrir des performances élevées et une architecture flexible. Dans une configuration à deux machines, chaque machine héberge des composants Frontend (FE) et Backend (BE).

- La première machine est désignée comme master, tandis que la seconde est un follower. La machine master contient un Frontend Node (FE) principal, qui gère les métadonnées, la planification des requêtes et la coordination globale des opérations. Elle héberge également un Backend Node (BE), responsable du stockage des données et de l'exécution des calculs.

- La machine follower, quant à elle, n'héberge pas de FE, mais contient un Backend Node (BE) secondaire, qui permet de répartir la charge de stockage et de calcul, tout en renforçant la tolérance aux pannes. Les données stockées et traitées par Doris sont visualisées avec Apache Superset, une plateforme de visualisation puissante et intuitive.

- Superset se connecte directement à Doris via une interface sécurisée et optimisée, permettant de créer des tableaux de bord interactifs et des visualisations personnalisées. Grâce à ses fonctionnalités avancées, Superset offre aux utilisateurs une vue claire et exploitable des données, facilitant ainsi la prise de décision et l'analyse approfondie des informations.

7. Configuration Détaillée

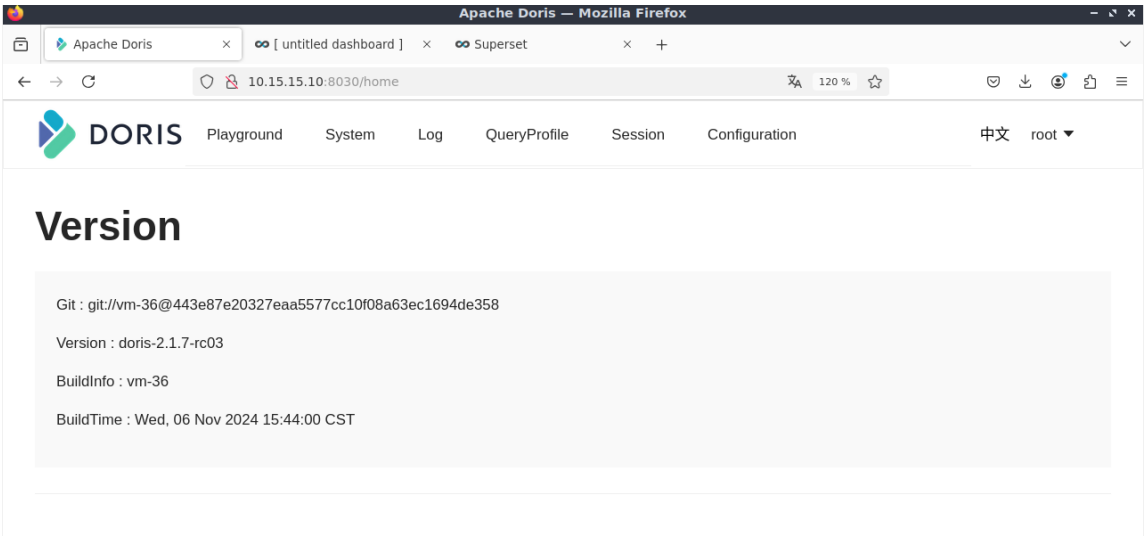
7.1 Configuration Frontend (fe.conf)

```
# Configuration des ports
http_port = 8030
rpc_port = 9020
query_port = 9030
edit_log_port = 9010

# Configuration JVM
JAVA_OPTS="-Xmx8192m -XX:+UseG1GC"

# Configuration système
meta_dir = /home/doris/metadata
priority_networks = 10.15.15.0/24
sys_log_level = INFO
```

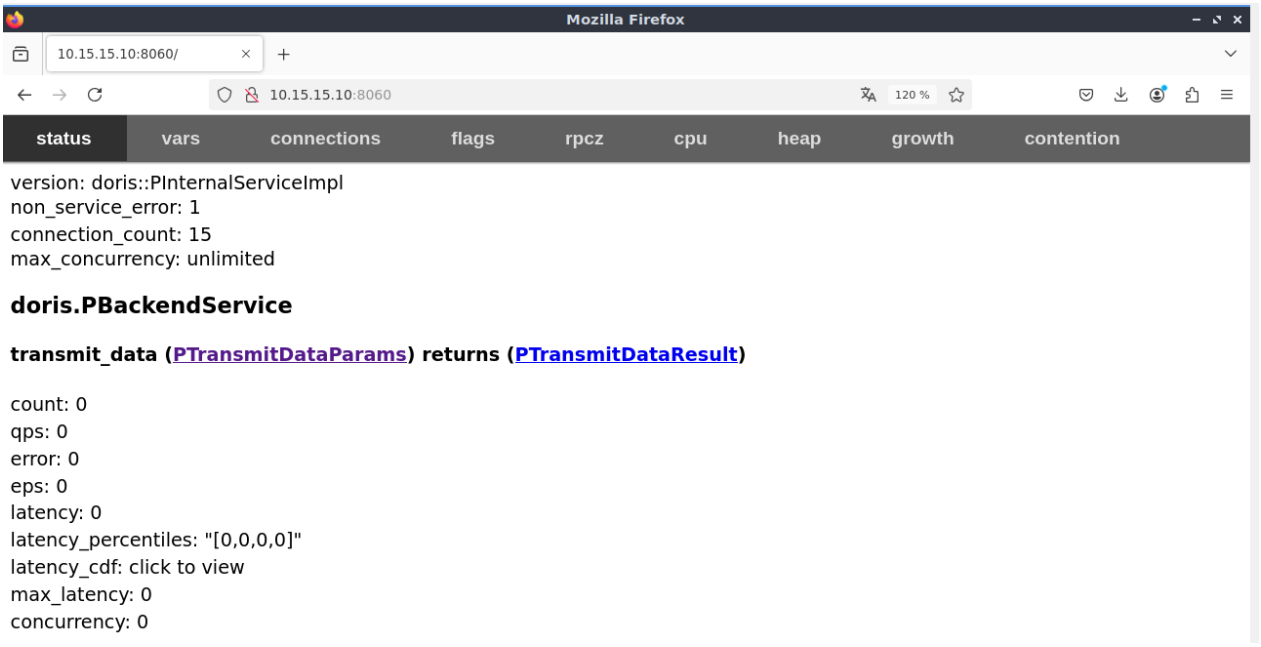
7.2 Configuration Backend(be.conf)

```
# Configuration des ports
be_port = 9060
webserver_port = 8040
heartbeat_service_port = 9050
brpc_port = 8060

# Configuration ressources
mem_limit = 4G
storage_root_path = /home/doris/stockage

# Configuration réseau
priority_networks = 10.15.15.0/24
```

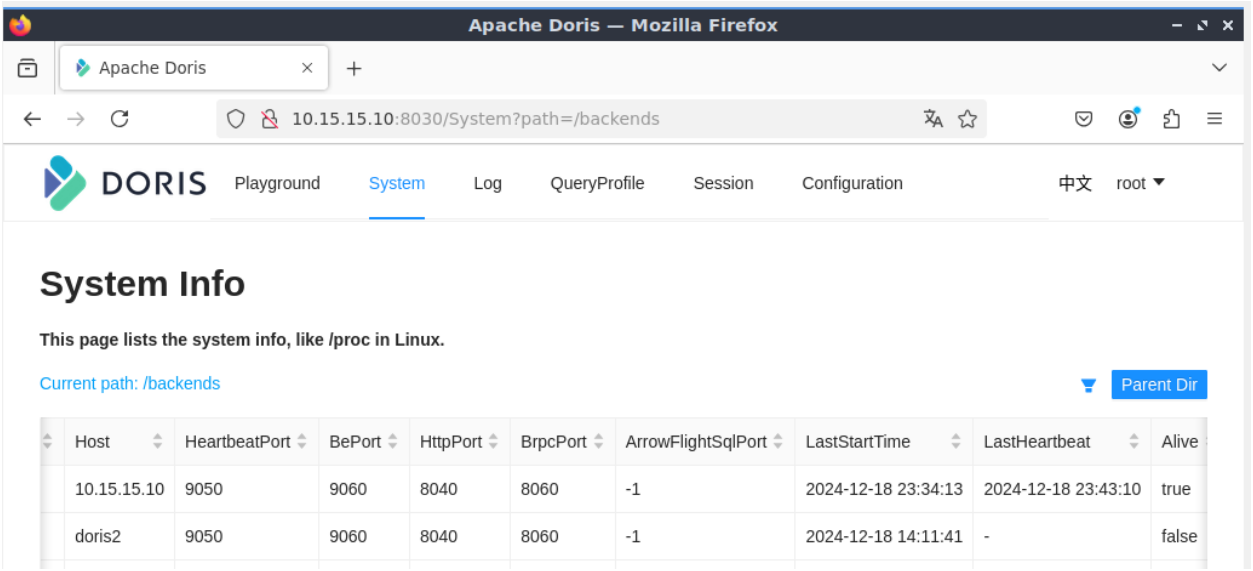
Illustration Pratique



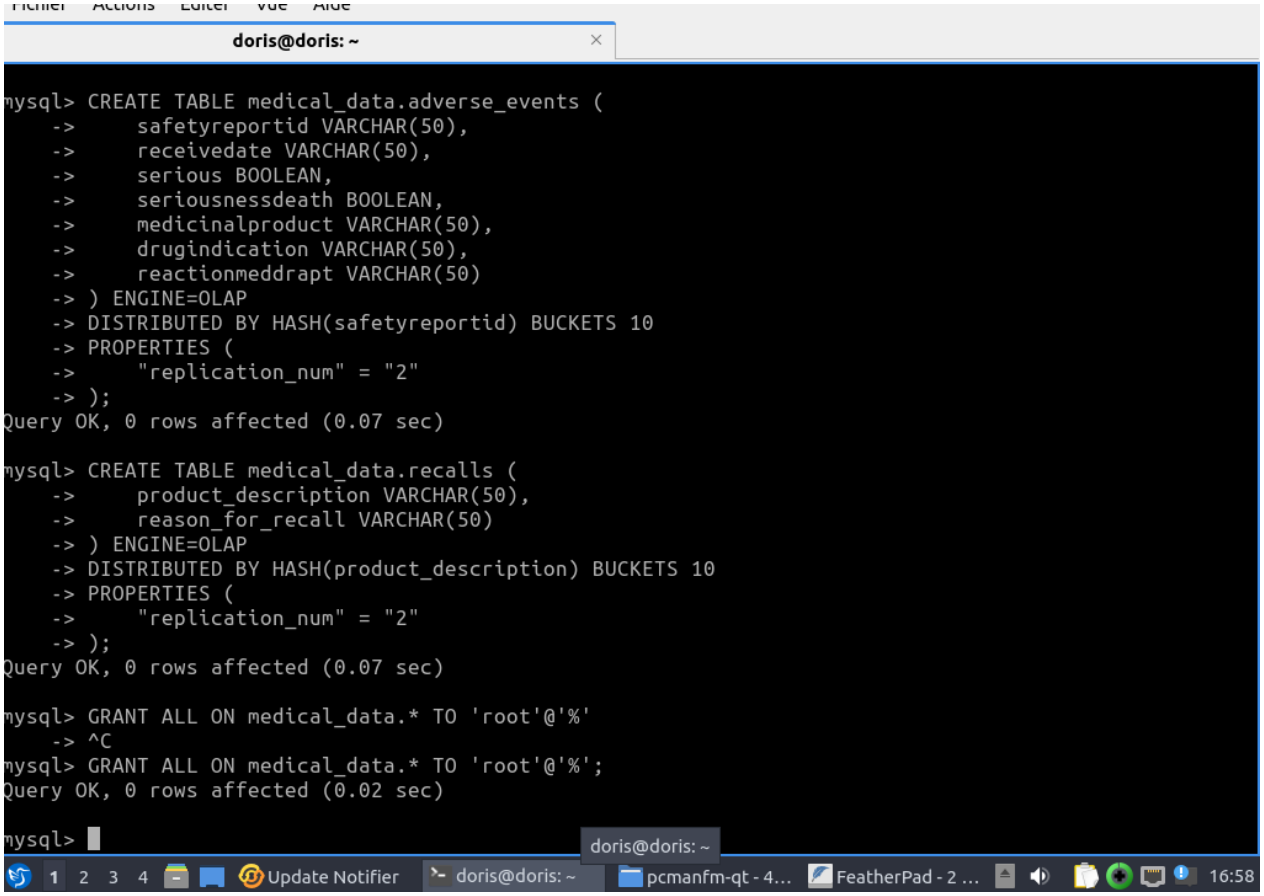
Puis notre partie backend :



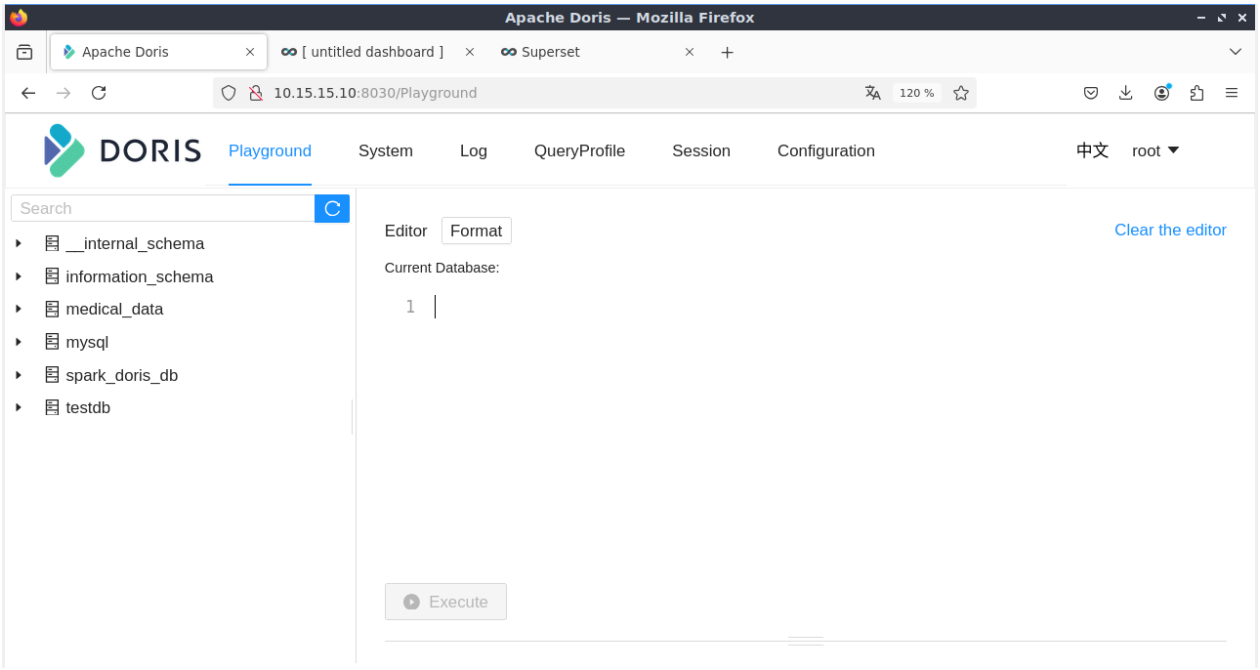
Afin d'assurer la haute disponibilité du backend nous avons ajoutée à doris un second backend



Puis nous avons créé notre dataset



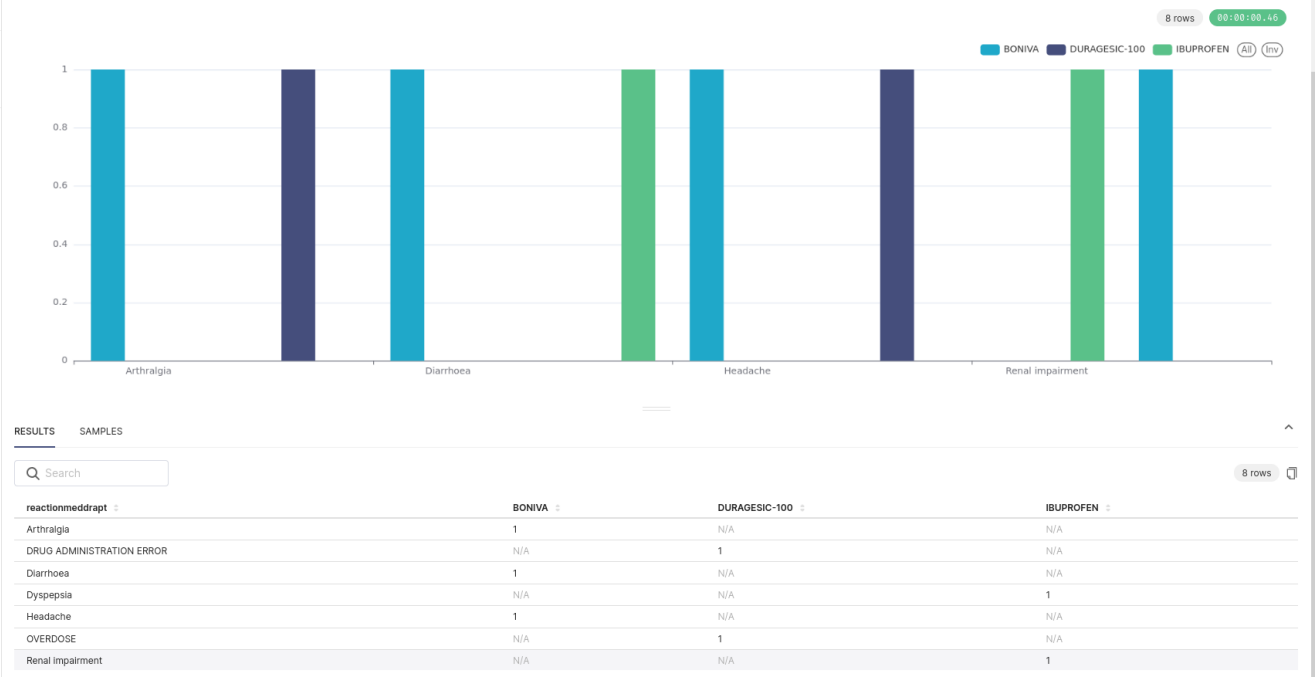
Cette interface montre bien les datasets par défaut et notre dataset crée :



Puis nous avons chargé les données avec spark :

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT * FROM medical_data.adverse_events;
+----------------+------------+--------+-----------------+-----------------+-----------------
-----------------+---------------------------+
| safetyreportid | receivedate | serious | seriousnessdeath | medicinalproduct | drugindication
                 | reactionmeddrapt          |
+----------------+------------+--------+-----------------+-----------------+-----------------
-----------------+---------------------------+
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Diarrhoea                 |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Vomiting                  |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Headache                  |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Arthralgia                |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Headache                  |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Arthralgia                |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Diarrhoea                 |
| 10003300       | 20140306    |       1 |                0 | BONIVA           | OSTEOPOROSIS
                 | Vomiting                  |
| 10003301       | 20140228    |       1 |                0 | IBUPROFEN        | PRODUCT USED FOR U
NKNOWN INDICATION | Renal impairment          |
| 10003301       | 20140228    |       1 |                0 | IBUPROFEN        | PRODUCT USED FOR U
NKNOWN INDICATION | Dyspepsia                 |
```

A travers ces données nous avons dégagée avec superset quelque visualisation :



RESULTS SAMPLES

Search

8 rows

reactionmeddrapt	BONIVA	DURAGESIC-100	IBUPROFEN
Arthralgia	1	N/A	N/A
DRUG ADMINISTRATION ERROR	N/A	1	N/A
Diarrhoea	1	N/A	N/A
Dyspepsia	N/A	N/A	1
Headache	1	N/A	N/A
OVERDOSE	N/A	1	N/A
Renal impairment	N/A	N/A	1

Comme effet secondiare arthralgia et headache est causés juste par **BONIVA**