

RAPPORT

TP : BIG DATA

Réalisé par :

BACHIRI Imane

Table des matières

1.	Introduction	3
1.1.	Contexte et importance du TP.....	3
1.2.	Objectifs.....	3
1.3.	Compétences acquises.....	4
2.	Vérification des composantes.....	4
2.1.	Docker.....	4
2.2.	Java	4
2.3.	Spark.....	4
3.	Mise en œuvre.....	5
3.1.	Démarrage des services Docker.....	5
3.2.	Développement & traitement des données.....	5
3.3.	Exécution du code.....	9





1. Introduction

1.1. Contexte et importance du TP

Dans un monde où les données volumineuses (Big Data) sont omniprésentes, la gestion efficace du stockage, du traitement et de l'analyse en temps réel est cruciale. Ce TP s'inscrit dans cette dynamique en mettant en pratique des outils et technologies clés du Big Data, tels que **Apache Kafka**, **Apache Spark**, **Docker** et **Minio**. L'objectif principal est de créer un pipeline de traitement de données en temps réel et de comprendre comment ces outils interagissent pour offrir une solution scalable et performante.

1.2. Objectifs

Ce TP vise à :

- **Installer et configurer l'environnement Big Data** 
 - Déployer un environnement de traitement distribué avec *Docker*.
 - Installer et configurer *Kafka* pour la gestion de flux de données.
 - Mettre en place *Spark* pour le traitement des données.
 - Configurer *Minio* comme solution de stockage en mode objet.
- **Produire et Consommer des données avec Kafka** 
 - Développer un *producer Kafka* en Python pour envoyer des données.
 - Récupérer ces données en *streaming* avec *Spark* ou *Kafka Streams*.
- **Transformer et enrichir les données** 
 - Stocker les données dans un *DataFrame Spark*.
 - Appliquer plusieurs transformations :
 - *Conversion des devises (USD → EUR).*
 - *Ajout du TimeZone.*
 - *Conversion des dates.*
 - *Filtrage des erreurs et des valeurs manquantes.*
- **Stocker et exploiter les données transformées** 
 - Écrire les données nettoyées et enrichies au format *Parquet* sur *Minio*.
 - Lire et exploiter ces données avec *Spark* pour validation.

1.3. Compétences acquises

À l'issue de ce TP, nous aurons acquis des compétences essentielles dans le domaine du Big Data :

- ✓ Déploiement et gestion d'un environnement Big Data avec *Docker*.
- ✓ Maîtrise des principes de *Kafka* et de la gestion des flux de données.
- ✓ Manipulation et transformation des données avec *Spark Streaming*.
- ✓ Stockage et récupération des données au format *Parquet* sur *Minio*.

2. Vérification des composantes

2.1. Docker

```
C:\Users\DELL>docker --version
Docker version 27.4.0, build bde2b89
```

```
C:\Users\DELL>docker-compose --version
Docker Compose version v2.31.0-desktop.2
```

2.2. Java

```
C:\Users\DELL>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

2.3. Spark

```
C:\Users\DELL>spark-shell --version
Welcome to

  ____ _
 / ___ \| | | |
/ /   \| |_| |
\ \   /| | | |
 \___/\_| |_| |
      |_|_|_|_|

 version 3.5.3

Using Scala version 2.12.18, Java HotSpot(TM) 64-Bit Server VM, 1.8.0_202
Branch HEAD
Compiled by user haejoon.lee on 2024-09-09T05:20:05Z
Revision 32232e9ed33bb16b93ad58cfde8b82e0f07c0970
Url https://github.com/apache/spark
Type --help for more information.
```

3. Mise en œuvre

3.1. Démarrage des services Docker

```
PS C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\docker-compose\minio> docker compose up -d
time="2025-02-09T11:58:51+01:00" level=warning msg="C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\docker-compose\minio\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[*] Running 17/12
✓ minio Pulled 20.1s
✓ mc Pulled 10.5s

time="2025-02-09T11:59:11+01:00" level=warning msg="a network with name spark_network exists but was not created for project \"minio\". \nSet 'external: true' to use an existing network"
[*] Running 2/2
✓ Container minio Started 3.0s
✓ Container mc Started 2.0s
```

<input type="checkbox"/>	▼	<input checked="" type="radio"/>	minio	-	-	-	0.21%	2 minutes		:	
<input type="checkbox"/>		<input checked="" type="radio"/>	minio	23dd7c6d5fd7	minio/minio	9000:9000	0.21%	2 minutes		:	
							Show all ports (2)				
<input type="checkbox"/>		<input type="radio"/>	mc	eccc70af4bde	minio/mc		0%	2 minutes		:	

➔ Vérifions que les conteneurs sont bien en cours d'exécution :

```
PS C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\docker-compose\minio> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
2da668cd17cf	konduktor/konduktor-data-generator:0.8	"java -cp @/app/jib-..."	2 minutes ago	Up About a minute	
62156afaca49	konduktor/konduktor-gateway:3.5.0	"java -cp @/app/jib-..."	2 minutes ago	Up 2 minutes (healthy)	0.0.0.0: konduktor-gateway
8888->8888/tcp	konduktor/konduktor-console:1.31.0	"/_cacert_entrypoint..."	2 minutes ago	Up 2 minutes (healthy)	0.0.0.0: minio-konduktor-console-1
1441102e3ec8	postgres:14	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes (healthy)	5432/tcp
8080->8080/tcp	postgres:14	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes (healthy)	minio-postgresql-2-1
22eeca0ad5fe	postgres:14	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes (healthy)	5432/tcp
d37d4bb4461e	docker.redpanda.com/redpandadata/redpanda:v24.1.6	"/entrypoint.sh redp..."	2 minutes ago	Up 2 minutes (healthy)	minio-postgresql-1
281aaf0b6fd8	2/tcp, 0.0.0.0:18081-18082->18081-18082/tcp, 9092/tcp, 0.0.0.0:19092->19092/tcp, 0.0.0.0:19644->9644/tcp		2 minutes ago	Up 2 minutes (healthy)	redpanda-0
555c52379fac	konduktor/konduktor-console-cortex:1.31.0	"/opt/konduktor/scri..."	2 minutes ago	Up 2 minutes (healthy)	8081-808
0/tcp, 9090/tcp					minio-konduktor-monitoring-1

```
PS C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\docker-compose\minio>
```

3.2. Développement & traitement des données

3.2.1. Producer Kafka (python)

```
producer = KafkaProducer(  
    bootstrap_servers=['localhost:19092'],  
    value_serializer=json_serializer  
)
```

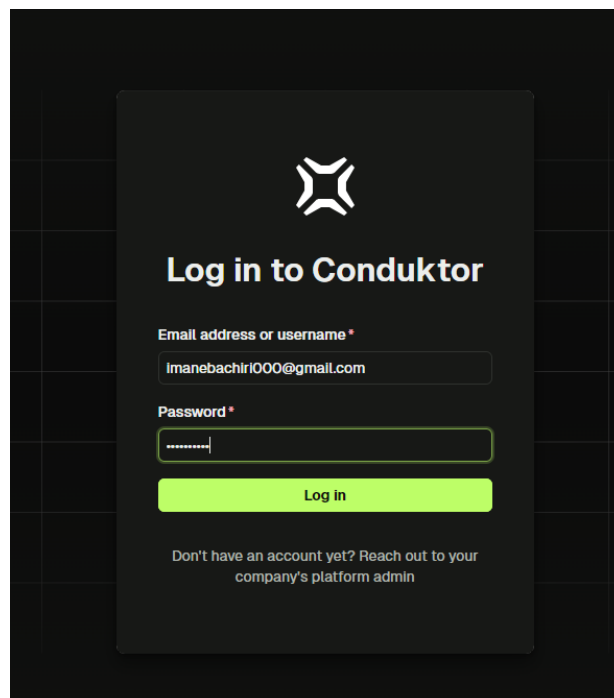
3.2.2. Application des transformations

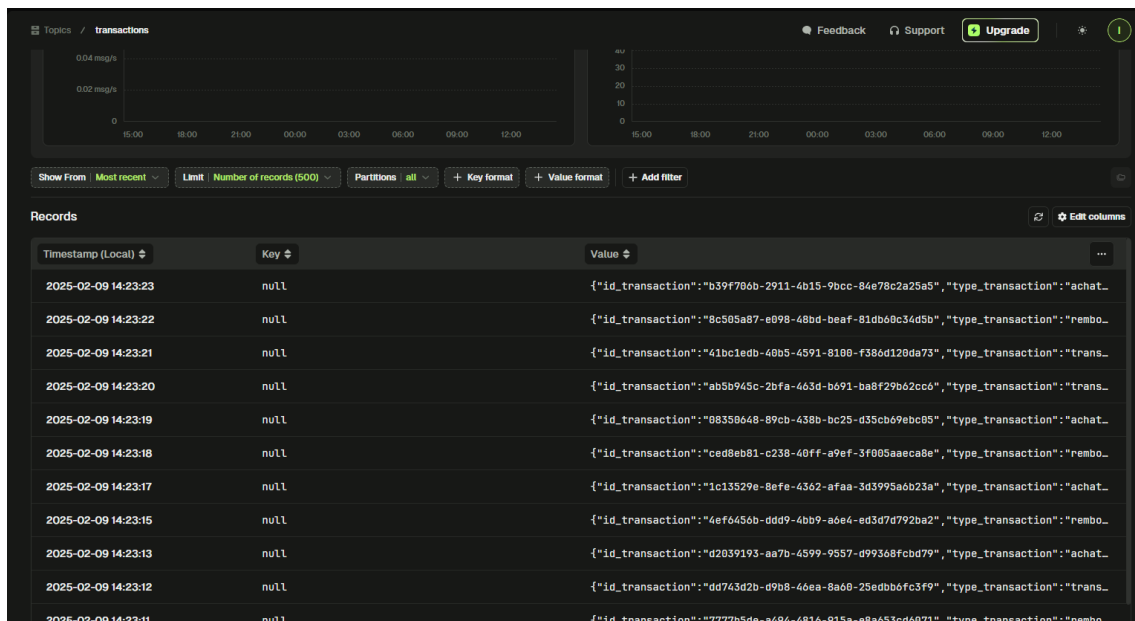
```
# Conversion USD en EUR  
if transaction_data["devise"] == "USD":  
    transaction_data["montant"] = round(transaction_data["montant"] * 0.85, 2)  
    transaction_data["devise"] = "EUR"  
  
# Vérifier et supprimer les transactions en erreur ou avec des valeurs None  
if transaction_data["moyen_paiement"] == "erreur" or \  
    None in [transaction_data["lieu"], transaction_data["adresse_utilisateur"]]:  
    return None
```

➔ Exécutons capteur.py :

```
PS C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\Python> python capteur.py
● Sent transaction: eef55152-a004-42e5-ada1-1d434021c73e
Sent transaction: 6fcec9e9-a48c-435e-be44-a780db0cc58c
Sent transaction: 8628c4de-f847-4161-b79b-a6b2c27be552
Sent transaction: 302712f7-bde5-4e74-adc9-82d23035b960
Sent transaction: 26a57460-ac08-46f0-bc02-91b3a058a803
Sent transaction: 0e578161-a6f7-4ef1-8e4d-fe242b2f6888
Sent transaction: 9b6fc49d-0028-4531-b32e-2e2d24d99bae
Sent transaction: 10181b87-2916-4901-95ce-b7e2a75bdd7e
Sent transaction: d6e4884d-3124-4b66-b261-7d216faa57a4
Sent transaction: 0bd90fc2-a7aa-42cc-815a-fcbcddeb31e28
Sent transaction: d038d19c-9dec-41e9-900e-9a06e47d77cf
Sent transaction: 0d7a6e9a-4ff5-451b-bee3-8f2e5cc8f3ea
Sent transaction: 60d723e8-bb01-40d9-9027-c62569066c7f
Sent transaction: 8ab4e0ba-3c77-4609-b071-141dd8d9008f
Sent transaction: ffbf502e-0526-48f6-9885-8ab1c3cfa1e7
Sent transaction: bc78aaa4-3ec0-465e-bc46-c5e6bdd95e63
Sent transaction: 1a76e362-c099-4136-8673-0900615cd013
Sent transaction: e3cb13e6-0c31-4278-9e0f-7f6b49b673e2
Sent transaction: 16d34e94-70fb-4f66-8c0b-d450fd0bfa82
Sent transaction: 7225a739-7b27-4720-87c9-772a4d1a87e9
Sent transaction: cf00b541-b4df-45b2-9303-b9bc21440f47
Sent transaction: ff2d2add-1926-43f4-9de6-59f5e2297b43
Sent transaction: ce16f703-f982-46e7-a410-bc8654c4b83c
Sent transaction: 163d6596-67b8-4b40-ae67-c0d392f3d48f
Sent transaction: 52cceadb-6902-4bc4-8b40-749b5dae9894
```

➔ Vérifions que les messages sur Conduktor Console





3.2.3. Traitement avec Spark Streaming

Pour récupérer les données via *Spark Streaming* ou *Kafka Stream* et les stocker dans un *DataFrame* Spark, voici les étapes détaillées que nous allons suivre :

- Configurer Spark Streaming avec Kafka

```
PS C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\docker-compose\minio> pip install pyspark kafka-python
Collecting pyspark
  Downloading pyspark-3.4.4.tar.gz (311.4 MB)
    311.4/311.4 MB 2.3 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Collecting kafka-python
  Downloading kafka-python-2.0.2-py2.py3-none-any.whl.metadata (7.8 kB)
Collecting py4j==0.10.9.7 (from pyspark)
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl.metadata (1.5 kB)
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
    200.5/200.5 kB 6.1 MB/s eta 0:00:00
  Downloading kafka_python-2.0.2-py2.py3-none-any.whl (246 kB)
    246.5/246.5 kB 5.0 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (pyproject.toml) ... done
  Created wheel for pyspark: filename=pyspark-3.4.4-py2.py3-none-any.whl size=311905498 sha256=6539acb339344ec6c5f0b7d4457e1eed8bc37f219498c37027b9e509258a106f
  Stored in directory: c:\users\de\appdata\local\pip\cache\wheels\4e\66\93\9eb1c49afb8a7fd2c4e393ad34e12b77db67bb4cc974c00e
Successfully built pyspark
Installing collected packages: py4j, kafka-python, pyspark
Successfully installed kafka-python-2.0.2 py4j-0.10.9.7 pyspark-3.4.4
```

- Créer une session Spark

```
# Création de la session Spark avec les dépendances S3
spark = SparkSession.builder \
    .appName("KafkaTransactionConsumer") \
    .master("local[*]") \
    .config("spark.jars.packages",
        "org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.3," +
        "org.apache.hadoop:hadoop-aws:3.3.4," +
        "com.amazonaws:aws-java-sdk-bundle:1.12.261") \
    .config("spark.hadoop.fs.s3a.endpoint", "http://localhost:9000") \
    .config("spark.hadoop.fs.s3a.access.key", "minio") \
    .config("spark.hadoop.fs.s3a.secret.key", "minio123") \
    .config("spark.hadoop.fs.s3a.path.style.access", "true") \
    .config("spark.hadoop.fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem") \
    .config("spark.streaming.stopGracefullyOnShutdown", "true") \
    .getOrCreate()
```

- Définir les schémas des données

```
▼ schema = StructType([
    StructField("id_transaction", StringType(), True),
    StructField("type_transaction", StringType(), True),
    StructField("montant", DoubleType(), True),
    StructField("devise", StringType(), True),
    StructField("date", StringType(), True),
    StructField("lieu", StringType(), True),
    StructField("moyen_paiement", StringType(), True),
    StructField("produit", StringType(), True),
    StructField("quantite", IntegerType(), True),
    StructField("prix_unitaire", DoubleType(), True),
    StructField("id_utilisateur", StringType(), True),
    StructField("nom_utilisateur", StringType(), True),
    StructField("adresse_utilisateur", StringType(), True),
    StructField("email_utilisateur", StringType(), True)
])
```

- Lire les données de Kafka en temps réel

```
# Lecture du flux Kafka
df = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "localhost:19092") \
    .option("subscribe", "transactions") \
    .option("startingOffsets", "earliest") \
    .load()
```

- Appliquer les transformations sur les données

```
# Transformation des données
parsed_df = df.select(
    from_json(col("value").cast("string"), schema).alias("data")
).select("data.*")

transformed_df = parsed_df \
    .withColumn("date", to_timestamp("date")) \
    .withColumn("timezone", lit("Europe/Paris")) \
    .filter(col("moyen_paiement") != "erreur") \
    .filter(col("adresse_utilisateur").isNotNull()) \
    .filter(col("lieu").isNotNull())
```


- Afficher ou écrire les données dans un DataFrame au format parquet dans Minio

```
# Écriture en streaming vers MinIO
query = transformed_df \
    .writeStream \
    .outputMode("append") \
    .format("parquet") \
    .option("path", "s3a://transactions/output") \
    .option("checkpointLocation", "s3a://transactions/checkpoints") \
    .start()

# Attente de la fin du traitement streaming
query.awaitTermination()
```

- Lire le fichier nouvellement créé sur Minio avec spark

```
def read_from_minio():
    df_minio = spark.read.parquet("s3a://transactions/output")
    print("Schema des données lues depuis MinIO:")
    df_minio.printSchema()
    print("\nAperçu des données:")
    df_minio.show(5)
    return df_minio

read_from_minio()
```

3.3. Exécution du code.

➔ Lancement du consumer Spark (consumer_spark.py) :

```

+-----+
| conf | modules | artifacts |
|-----+-----+-----+
|      | number | search | dwnlded | evicted | | number | dwnlded |
+-----+-----+-----+
| default | 11 | 3 | 3 | 0 | | 11 | 3 |
+-----+-----+-----+

:: retrieving :: org.apache.spark#spark-submit-parent-5ed653a6-f006-41af-95b1-f5e7c776842f
  confs: [default]
  3 artifacts copied, 8 already retrieved (2723kB/62ms)
25/02/09 21:59:31 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
root
 |-- id_transaction: string (nullable = true)
 |-- type_transaction: string (nullable = true)
 |-- montant: double (nullable = true)
 |-- devise: string (nullable = true)
 |-- date: timestamp (nullable = true)
 |-- lieu: string (nullable = true)
 |-- moyen_paiement: string (nullable = true)
 |-- produit: string (nullable = true)
 |-- quantite: integer (nullable = true)
 |-- prix_unitaire: double (nullable = true)
 |-- id_utilisateur: string (nullable = true)
 |-- nom_utilisateur: string (nullable = true)
 |-- adresse_utilisateur: string (nullable = true)
 |-- email_utilisateur: string (nullable = true)
 |-- timezone: string (nullable = false)

```

➔ Exécution du producteur de données (capteur.py) :

```
PS C:\Users\DELL\Downloads\tp_BigDATA\tp_BigDATA\Python> python capteur.py
● Sent transaction: 3bc5bc3b-8a02-4606-876e-3ada22bfffefe
Sent transaction: ed8502dd-01fb-432f-b494-5aefece8a0aa
Sent transaction: 5546a546-e25a-40b9-9b94-c5a117edff70
Sent transaction: 668b42fb-b078-474a-ba24-ee8e2aac0aa
Sent transaction: 85452338-0087-45a2-b25d-fa3cbe29c14b
Sent transaction: a4d93f42-a1ed-4da9-8249-5d85496b79e0
Sent transaction: 63e4ddba-2ff2-464a-8a07-fd1e43713ee8
Sent transaction: 895c0b8c-c974-4ada-9e62-897240c75fe1
Sent transaction: 3e72e248-0b50-4ee5-9c32-6ac9ed6dba31
Sent transaction: af7f7bae-78ae-48b4-80f8-77581c997dfd
Sent transaction: 8ed790f0-d09f-4888-be15-ae0041ae99d6
Sent transaction: c7e4d287-c492-4035-acf8-6ae5df6b9964
Sent transaction: 683c0858-e02f-4f4d-b33e-e211a5f955c9
Sent transaction: 3907c1bd-65bb-4597-9d01-aa0f8881718a
Sent transaction: 52b6c14a-b4cf-448c-b4a1-3c06656e552f
Sent transaction: 2863e02e-7dbb-4700-a502-7bf67b00d3c5
Sent transaction: 1b753e21-2fad-4182-b6cf-60f851fd8a75
```

La communication entre le producteur et le consumer s'est établie correctement, permettant le traitement en temps réel des données de transaction via Spark Streaming.

```
25/02/09 22:45:43 WARN MetricsConfig: Cannot locate configuration: tried hadoop-metrics2-s3a-file-system.properties,hadoop-metrics2.properties
25/02/09 22:45:52 WARN AdminClientConfig: These configurations '[key.deserializer, value.deserializer, enable.auto.commit, max.poll.records, auto.offset.reset]' were supplied but are not used yet.
[Stage 1:>                                (0 + 1) / 1]
```

Schema des données lues depuis MinIO:

```
root
|-- id_transaction: string (nullable = true)
|-- type_transaction: string (nullable = true)
|-- montant: double (nullable = true)
|-- devise: string (nullable = true)
|-- date: timestamp (nullable = true)
|-- lieu: string (nullable = true)
|-- moyen_paiement: string (nullable = true)
|-- produit: string (nullable = true)
|-- quantite: integer (nullable = true)
|-- prix_unitaire: double (nullable = true)
|-- id_utilisateur: string (nullable = true)
|-- nom_utilisateur: string (nullable = true)
|-- adresse_utilisateur: string (nullable = true)
|-- email_utilisateur: string (nullable = true)
|-- timezone: string (nullable = false)
```

Aperçu des données:

id_transaction	type_transaction	montant	devise	date	lieu	moyen_paiement	produit	quantite	prix_unitaire	id_utilisateur	nom_utilisateur	adresse_utilisateur	email_utilisateur	timezone
ee5152-a004-42e...	remboursement	157.45	EUR	2025-02-09 14:21:...	None, None	carte_de_credit	Produit25	9	172.01	User856	Utilisateur191	735 Rue de Paris,...	utilisateur372@ex...	Europe/Paris
6fcec9e9-a48c-435...	transfert	407.39	EUR	2025-02-09 14:21:...	Rue Auguste Delau...	carte_de_credit	Produit14	5	33.77	User588	Utilisateur568	238 Rue de Paris,...	utilisateur657@ex...	Europe/Paris
8628c4de-f847-416...	remboursement	934.76	EUR	2025-02-09 14:21:...	Rue Auguste Delau...	especes	Produit70	9	150.45	User945	Utilisateur884	977 Rue de la Rép...	utilisateur494@ex...	Europe/Paris
302712f7-bde5-4e7...	achat	124.41	EUR	2025-02-09 14:21:...	Rue de la Républi...	carte_de_credit	Produit34	4	112.89	User299	Utilisateur74	207 Rue de la Rép...	utilisateur357@ex...	Europe/Paris
26a57460-ac08-46f...	transfert	525.22	EUR	2025-02-09 14:21:...	None, None	especes	Produit81	3	118.46	User20	Utilisateur144	137 None, Lyon	utilisateur50@exa...	Europe/Paris

only showing top 5 rows

Passons maintenant a la plateforme de Minio : <http://localhost:9001> Pour vérifier si les données sont bien enregistrés en format paquet

Username : Minio

Password : Minio123

➔ *Bucket : transactions*

transactions

Created on: Sun, Feb 09 2025 21:29:58 (GMT+1) Access: PUBLIC 90.0 B - 4 Objects

Rewind ↶

Refresh ↺

Upload ↗

transactions

Create new path ↗

<input type="checkbox"/>	Name	Last Modified	Size
<input type="checkbox"/>	checkpoints		-
<input type="checkbox"/>	output		-
<input type="checkbox"/>	processed		-

➔ *Folder : checkpoint*

transactions

Created on: Sun, Feb 09 2025 21:29:58 (GMT+1) Access: PUBLIC 90.0 B - 4 Objects

Rewind ↶

Refresh ↺

Upload ↗

transactions / checkpoints

Create new path ↗

<input type="checkbox"/>	Name	Last Modified	Size
<input type="checkbox"/>	commits		-
<input type="checkbox"/>	metadata	Today, 22:28	45.0 B
<input type="checkbox"/>	offsets		-
<input type="checkbox"/>	sources		-

➔ *Folder : output*

transactions

Created on: Sun, Feb 09 2025 21:29:58 (GMT+1) Access: PUBLIC 146.1 KiB - 55 Objects

Rewind ↶

Refresh ↺

Upload ↗

transactions / output

Create new path ↗

<input type="checkbox"/>	Name	Last Modified	Size
<input type="checkbox"/>	_spark_metadata		-
<input type="checkbox"/>	part-00000-09532006-ce23-419e-b320-76d90e41fc15-c000.snappy.parquet	Today, 22:34	5.3 KiB
<input type="checkbox"/>	part-00000-0c270944-4053-404e-b553-2520defeac31-c000.snappy.parquet	Today, 22:47	5.3 KiB
<input type="checkbox"/>	part-00000-1498e39d-d5c6-4403-bfd-d-2efbe44a32b8-c000.snappy.parquet	Today, 22:32	5.4 KiB
<input type="checkbox"/>	part-00000-256692d6-fc0f-4727-bf8c-9a5cb8f814b3-c000.snappy.parquet	Today, 22:34	5.5 KiB
<input type="checkbox"/>	part-00000-40d6ce5d-08ae-4dd2-938c-3c440d83b6ae-c000.snappy.parquet	Today, 22:46	5.3 KiB
<input type="checkbox"/>	part-00000-4684cc39-476a-4f5e-b413-cc401d56929-c000.snappy.parquet	Today, 22:46	5.8 KiB
<input type="checkbox"/>	part-00000-520b1faa-9884-44f9-8104-6f86e9cb47fd-c000.snappy.parquet	Today, 22:33	5.7 KiB
<input type="checkbox"/>	part-00000-5388e77b-1ed0-4166-a538-7f86103f8c40-c000.snappy.parquet	Today, 22:33	6.2 KiB
<input type="checkbox"/>	part-00000-6666854f-03d4-4adf-bff0-3916bcd1a1b8-c000.snappy.parquet	Today, 22:47	5.2 KiB
<input type="checkbox"/>	part-00000-6ba6a0f1-92d6-4fd4-b7ad-a84a8e9a51c9-c000.snappy.parquet	Today, 22:33	5.5 KiB
<input type="checkbox"/>	part-00000-7268fc30-43fb-4688-97c4-19599e86a4a6-c000.snappy.parquet	Today, 22:46	6.1 KiB
<input type="checkbox"/>	part-00000-78b0fc7a-b033-438f-b207-5802ca470ed7-c000.snappy.parquet	Today, 22:47	5.2 KiB
<input type="checkbox"/>	part-00000-8a2db9e7-5c73-4f46-95be-ff9a52eb1b33-c000.snappy.parquet	Today, 22:47	5.2 KiB

FIN