1. INTRODUCTION

Ce projet a pour objectif de construire une plateforme analytique Data Lakehouse pour un distributeur de vélos. L'entreprise souhaite moderniser son infrastructure de données pour mieux piloter son activité après la forte augmentation des ventes pendant la pandémie.

Contexte

Le client dispose de **4 fichiers CSV** contenant :

- **bikes.csv** : catalogue des vélos (modèle, catégorie, marque, prix)
- **bikeshops.csv** : liste des magasins (nom, ville, région, pays)
- **orders.csv** : commandes clients (date, client, produit, quantité, prix)
- **customers.csv**: informations clients .

.

Objectifs du projet

Les objectifs principaux du projet sont les suivants.

- Concevoir et implémenter un pipeline ETL Spark permettant de transformer les fichiers CSV sources en tables Parquet.
- Construire un DataMart en étoile composé de plusieurs tables de dimensions et d'une table de faits des ventes.
- Proposer des requêtes analytiques permettant d'explorer les ventes selon différents axes (produit, magasin, client, temps, promotions).
- Mettre en place une observabilité des jobs Spark en exposant des métriques collectées par Prometheus et visualisées dans Grafana.

Technologies utilisées

- **Apache Spark 3.x** avec PySpark
- **Docker** pour conteneurisation
- **HDFS** pour le stockage distribué
- **Parquet** comme format
- **Prometheus** pour la collecte de métriques
- **Grafana** pour construire les tableaux de bord d'observabilité.

2. ARCHITECTURE TECHNIQUE

2.1 Schéma d'architecture global

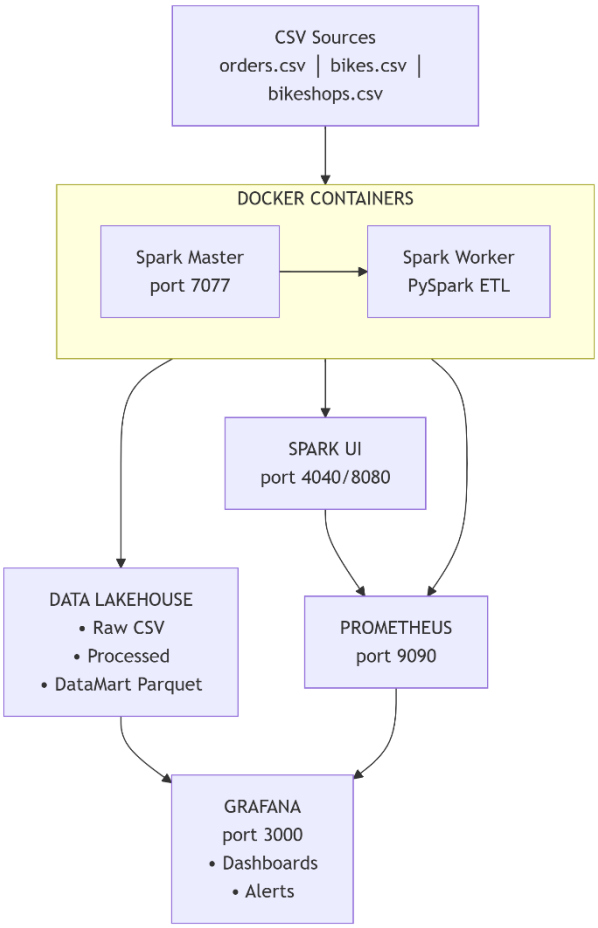


Figure 1 : Architecture globale du projet Data Lakehouse

1. COUCHE DONNÉES

- **Sources** : 4 fichiers CSV (orders.csv, bikes.csv, bikeshops.csv,customers.csv)
- **Zone Raw** : Données brutes en HDFS
- **Zone Processed** : Données nettoyées en Parquet
- **DataMart** : Schéma en étoile pour l'analytique

2. COUCHE TRAITEMENT

- **Docker Containers** : Environnement isolé et portable
- **Spark Master** : Orchestrateur des jobs (port 7077)
- **Spark Workers** : Exécution distribuée PySpark ETL
- **PySpark** : Transformation, nettoyage, jointures

3. COUCHE OBSERVABILITÉ

- **Spark UI** : Monitoring temps réel (ports 4040/8080)
- **Prometheus** : Collecte métriques (port 9090)
- **Grafana** : Dashboards

3.Pipeline ETL Spark et construction du DataMart en étoile

Fichier : etl_velos_final

Le pipeline ETL, implémenté en PySpark et exécuté dans un conteneur Spark, charge les fichiers CSV depuis la zone Raw. Il effectue des contrôles de qualité (dédoublonnage, gestion des valeurs manquantes, vérification des clés de référence) et des transformations d'enrichissement (jointures des tables, calculs de montants..)Les données nettoyées sont ensuite écrites au format Parquet. Le DataMart Ventes, conçu selon un schéma en étoile, comprend des tables de dimensions (bikes, customers, bikeshops, date) et une table de faits (orders) permettant des analyses multidimensionnelles performantes des ventes.

	Name	Container ID	Image	Port(s)	Actions
○	mystifying_dew(	be5a0a37f92c	hello-world		
○	welcome-to-doc	0dced3913e17	docker/wel(	8088:80	
●	spark-master	b1f057285cda	bde2020/s(	7077:7077 Show all ports (2)	
●	spark-worker	30b0cd29c654	bde2020/s(	8081:8081	
●	spark-velos	e43688c8bd72	apache/spa	4040:4040 Show all ports (2)	
●	grafana	17ed56de6396	grafana/gra	3000:3000	

Spark Master at spark://b1f057285cda:7077

URL: spark://b1f057285cda:7077
Alive Workers: 1
Cores in use: 2 Total, 0 Used
Memory in use: 2.8 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 4 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20251217123824-172.18.0.3-45059	172.18.0.3:45059	ALIVE	2 (0 Used)	2.8 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Completed Applications (4)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20251217130705-0003	VentesVelosFinal	2	1024.0 MiB		2025/12/17 13:07:05	root	FINISHED	13 s
app-20251217130134-0002	VentesVelosFinal	2	1024.0 MiB		2025/12/17 13:01:34	root	FINISHED	14 s
app-20251217125152-0001	VentesVelos	2	1024.0 MiB		2025/12/17 12:51:52	root	FINISHED	15 s
app-20251217125023-0000	VentesVelos	2	1024.0 MiB		2025/12/17 12:50:23	root	FINISHED	22 s

```
=== VERIFICATION TERMINEE ===
25/12/17 14:15:39 INFO SparkUI: Stopped Spark web UI at http://30b0cd29c654:4040
25/12/17 14:15:39 INFO StandaloneSchedulerBackend: Shutting down all executors
25/12/17 14:15:39 INFO CoarseGrainedSchedulerBackend$DriverEndpoint: Asking each executor to shut down
25/12/17 14:15:39 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/12/17 14:15:39 INFO MemoryStore: MemoryStore cleared
25/12/17 14:15:39 INFO BlockManager: BlockManager stopped
25/12/17 14:15:39 INFO BlockManagerMaster: BlockManagerMaster stopped
25/12/17 14:15:39 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/12/17 14:15:39 INFO SparkContext: Successfully stopped SparkContext
25/12/17 14:15:40 INFO ShutdownHookManager: Shutdown hook called
25/12/17 14:15:40 INFO ShutdownHookManager: Deleting directory /tmp/spark-f4a4261d-99f8-43e6-9c41-35f1ee064cf7
25/12/17 14:15:40 INFO ShutdownHookManager: Deleting directory /tmp/spark-c34c1853-37bd-4ea4-8ba7-2a892edec698
25/12/17 14:15:40 INFO ShutdownHookManager: Deleting directory /tmp/spark-c34c1853-37bd-4ea4-8ba7-2a892edec698/pyspark-28ac8261-68ed-4c82-ab54-d136b
376686b
PS C:\Users\MERIEM\spark-velos> docker cp spark-worker:/tmp/factorders_final C:\Users\MERIEM\spark-velos\
Successfully copied 56.3kB to C:\Users\MERIEM\spark-velos\
PS C:\Users\MERIEM\spark-velos> docker exec spark-worker ls -la /tmp/factorders_final
total 68
drwxr-xr-x    2 root     root          4096 Dec 17 14:10 .
drwxrwxrwt    1 root     root          4096 Dec 17 14:15 ..
-rw-r--r--    1 root     root             8 Dec 17 14:10 ._SUCCESS.crc
-rw-r--r--    1 root     root           412 Dec 17 14:10 .part-00000-2604697e-2c67-4f4e-9d26-077bbf0b04d6-c000.snappy.parquet.crc
-rw-r--r--    1 root     root             0 Dec 17 14:10 _SUCCESS
-rw-r--r--    1 root     root         51505 Dec 17 14:10 part-00000-2604697e-2c67-4f4e-9d26-077bbf0b04d6-c000.snappy.parquet
PS C:\Users\MERIEM\spark-velos>
```

4. Analyses et requêtes analytiques

Fichier : sql_queries

```
+---------------------+----------+-------------+
|bike_model           |sum(amount)|sum(quantity)|
+---------------------+----------+-------------+
|F-Si 2               |269860.0   |131          |
|CAAD12 Red           |377600.0   |118          |
|Slice Ultegra        |315900.0   |117          |
|CAAD8 Sora           |119480.0   |116          |
|Beast of the East 1  |307470.0   |111          |
|CAAD8 105            |155100.0   |110          |
|Supersix Evo Ultegra 3|348800.0   |109          |
|Trail 5              |88020.0    |108          |
|Scalpel-Si 5         |336000.0   |105          |
|Trail 4              |101920.0   |104          |
+---------------------+----------+-------------+
only showing top 10 rows
```

```
+----------+-----------+-------------+
|order_year|sum(amount)|sum(quantity)|
+----------+-----------+-------------+
|      2011|  4296565.0|         1262|
|      2012|  4914795.0|         1357|
|      2013|  6806500.0|         1924|
|      2014|  4652315.0|         1317|
|      2015|  6365790.0|         1740|
+----------+-----------+-------------+


Top 10 des modeles les plus vendus :
```

```
25/12/17 14:15:34 INFO CodeGenerator: Code generated in 8.35433
+----------+-----------+-------------+
|order_year|sum(amount)|sum(quantity)|
+----------+-----------+-------------+
|      2011|  4296565.0|         1262|
|      2012|  4914795.0|         1357|
|      2013|  6806500.0|         1924|
|      2014|  4652315.0|         1317|
|      2015|  6365790.0|         1740|
+----------+-----------+-------------+


Top 10 des modeles les plus vendus :
```

```
25/12/17 14:15:27 INFO TaskSchedulerImpl: Killing all running tasks in stage 11: Stage fin
25/12/17 14:15:27 INFO DAGScheduler: Job 6 finished: collect at /tmp/check_results.py:21,
Prix moyen : 3555.43485675

Ventes par annee :
25/12/17 14:15:25 INFO TaskSche
25/12/17 14:15:25 INFO DAGSched
Montant total : 27035965.0
25/12/17 14:15:25 INFO FileSour
25/12/17 14:15:26 INFO TaskS
25/12/17 14:15:26 INFO DAGSc
Quantite totale : 7600
25/12/17 14:15:26 INFO FileS
25/12/17 14:15:26 INFO TaskSchedulerImpl: Killi
25/12/17 14:15:26 INFO DAGScheduler: Job 5 fini
Nombre moyen par commande : 1.29604365621
25/12/17 14:15:26 INFO FileSourceStrategy: Prun
```

```
25/12/17 14:15:34 INFO CodeGenerator: Code generat
+----------+-----------+-------------+
|order_year|sum(amount)|sum(quantity)|
+----------+-----------+-------------+
|      2011|  4296565.0|         1262|
|      2012|  4914795.0|         1357|
|      2013|  6806500.0|         1924|
|      2014|  4652315.0|         1317|
|      2015|  6365790.0|         1740|
+----------+-----------+-------------+
```

```
25/12/17 14:30:30 INFO CodeGenerator: Code ge
+--------------------+--------+------+
|          bike_model|ca_euros|unites|
+--------------------+--------+------+
|Habit Hi-Mod Blac...|808500.0|    66|
|     F-Si Black Inc.|794490.0|    71|
|Scalpel-Si Black ...|780190.0|    61|
|Supersix Evo Blac...|703450.0|    55|
|Synapse Hi-Mod Di...|652120.0|    68|
|     Scalpel-Si Race|643260.0|    71|
|Supersix Evo Hi-M...|639600.0|    60|
|     Jekyll Carbon 1|615230.0|    77|
| Scalpel-Si Hi-Mod 1|522200.0|    70|
|Scalpel 29 Carbon...|485640.0|    76|
+--------------------+--------+------+
```

```
+--------------+------------+-----------------+---------+
|ca_total_euros|nb_commandes|    panier_moyen|nb_lignes|
+--------------+------------+-----------------+---------+
|   2.7035965E7|         773|4610.49880627558|     5864|
+--------------+------------+-----------------+---------+
```

```
+--------------------+----------+---------+
|          bike_model|prix_moyen|nb_ventes|
+--------------------+----------+---------+
|Supersix Evo Blac...|   12790.0|       45|
|Scalpel-Si Black ...|   12790.0|       48|
|Habit Hi-Mod Blac...|   12250.0|       46|
|     F-Si Black Inc.|   11190.0|       54|
|Supersix Evo Hi-M...|   10660.0|       50|
+--------------------+----------+---------+
```

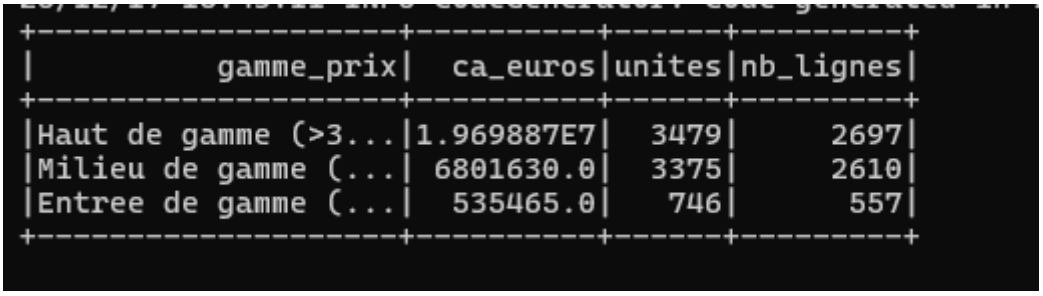
order_date	ca_euros	unites
2013-12-03	405645.0	111
2015-12-03	347875.0	71
2013-05-02	299045.0	64
2015-04-06	289500.0	79
2014-10-06	272190.0	60
2012-06-11	263265.0	51
2011-12-04	262890.0	71
2013-01-11	255400.0	82
2013-04-09	254005.0	76
2015-05-03	245780.0	65

annee	bike_model	unites
2011	CAAD8 Sora	41
2012	Trail 5	30
2013	Trail 4	37
2014	F-Si 1	28
2014	Slice Ultegra	28
2015	F-Si 2	40

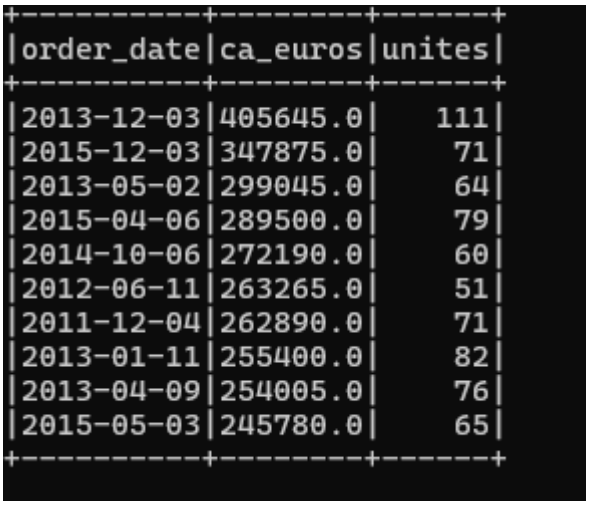
nb_modeles	nb_bikes_ids
97	97

ca_moyen_par_ligne	qte_moyenne_par_ligne
4610.49880627558	1.296043656207367

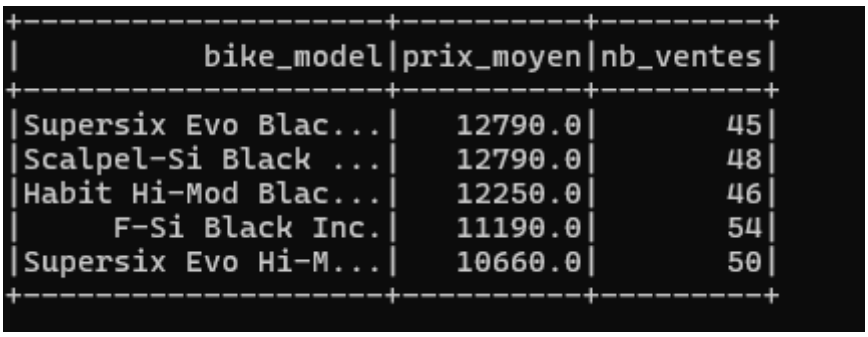
annee	bike_model	unites
2011	CAAD8 Sora	41
2012	Trail 5	30
2013	Trail 4	37
2014	F-Si 1	28
2014	Slice Ultegra	28
2015	F-Si 2	40



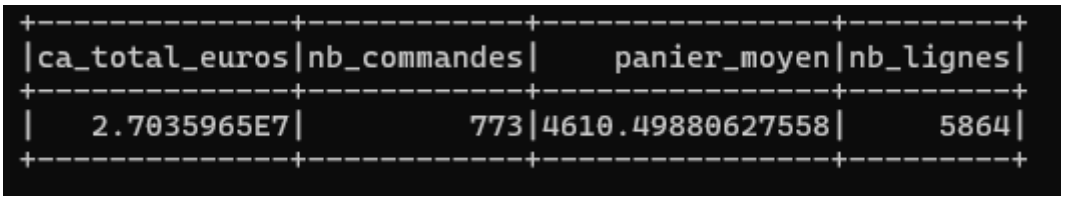
gamme_prix	ca_euros	unites	nb_lignes
Haut de gamme (>3...	1.969887E7	3479	2697
Milieu de gamme (...	6801630.0	3375	2610
Entree de gamme (...	535465.0	746	557



order_date	ca_euros	unites
2013-12-03	405645.0	111
2015-12-03	347875.0	71
2013-05-02	299045.0	64
2015-04-06	289500.0	79
2014-10-06	272190.0	60
2012-06-11	263265.0	51
2011-12-04	262890.0	71
2013-01-11	255400.0	82
2013-04-09	254005.0	76
2015-05-03	245780.0	65



bike_model	prix_moyen	nb_ventes
Supersix Evo Blac...	12790.0	45
Scalpel-Si Black ...	12790.0	48
Habit Hi-Mod Blac...	12250.0	46
F-Si Black Inc.	11190.0	54
Supersix Evo Hi-M...	10660.0	50

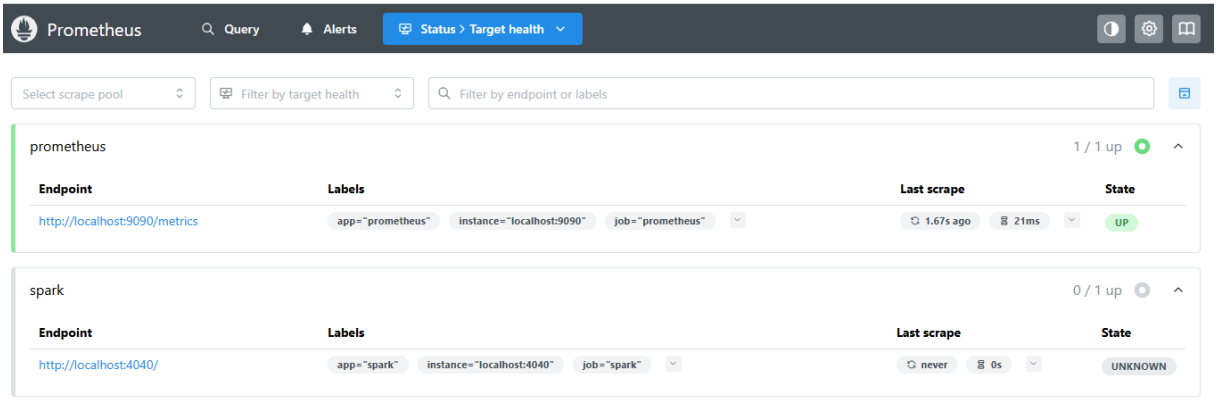


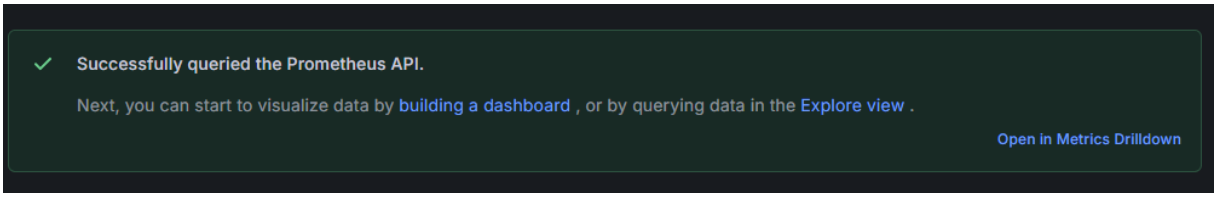
ca_total_euros	nb_commandes	panier_moyen	nb_lignes
2.7035965E7	773	4610.49880627558	5864

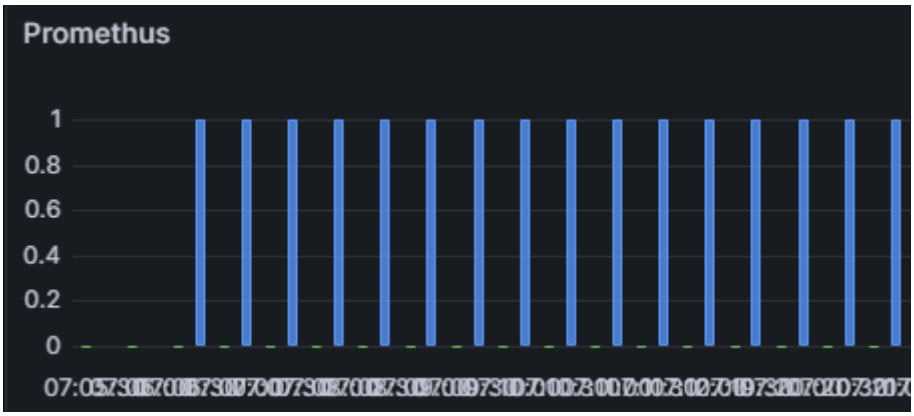
7. Observabilité

Le pipeline ETL intègre une solution complète d'observabilité basée sur la stack Prometheus/Grafana : les métriques Spark (statut et durée des jobs, volumes de données, erreurs, utilisation des ressources) sont exposées via un endpoint HTTP configuré avec PrometheusServlet, puis collectées périodiquement par Prometheus qui les stocke sous forme de séries temporelles. Enfin, Grafana visualise ces métriques via des dashboards dédiés, permettant de monitorer en temps réel les performances des jobs ETL, le volume de données traitées, les lignes rejetées lors des contrôles qualité, ainsi que l'utilisation CPU/mémoire des exécuteurs et la fraîcheur des données,

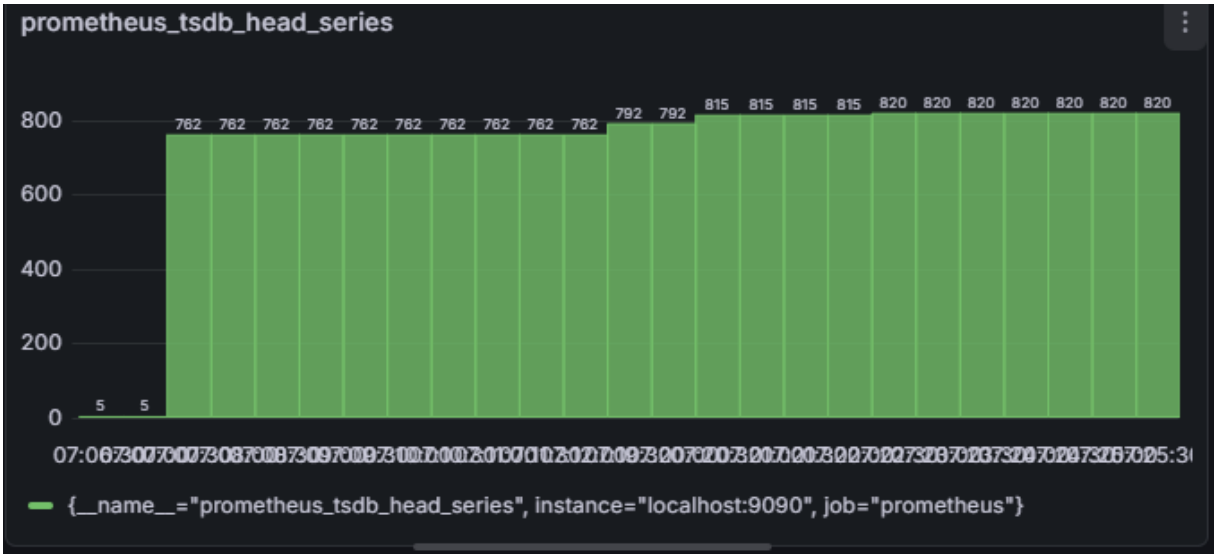
assurant ainsi une traçabilité complète du pipeline.

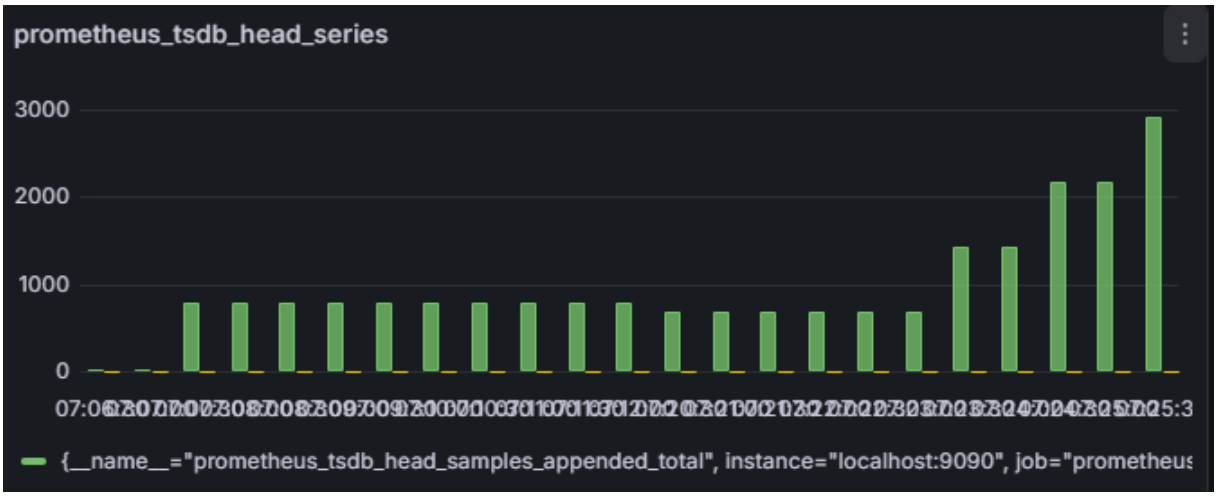





Promethus
1
0.8
0.6
0.4
0.2
0
07:0


spark
10
8
6
4
2
0
{__name__="scrape_duration_seconds", instance="172.17.0.2:4040", job="spark"}


prometheus_tsdb_head_series
800
600
400
200
0
762 762 762 762 762 762 762 762 762 762
792 792
815 815 815 815
820 820 820 820 820 820 820
5 5
{__name__="prometheus_tsdb_head_series", instance="localhost:9090", job="prometheus"}


prometheus_tsdb_head_series
3000
2000
1000
0
{__name__="prometheus_tsdb_head_samples_appended_total", instance="localhost:9090", job="prometheus