ПРОТОКОЛ
внедрения и тестирования Kafka Connect Iceberg Sink Connector
в тестовой среде, развернутой с использованием инструмента dpd

г. Краснодар
«__» _____ 2025 г.

1. Введение

Настоящий протокол составлен по результатам выполнения рабочей задачи в рамках технологической инфраструктуры ПАО «Магнит».
Исполнитель: _____ (Ф.И.О., должность)
Организация: Публичное акционерное общество «Магнит» (ПАО «Магнит»)

2. Постановка задачи

В рамках стратегии развития аналитической платформы и внедрения современных технологий хранения и обработки больших данных, перед Исполнителем была поставлена задача по исследованию и тестированию интеграции потоковых данных из Apache Kafka в озеро данных на базе Apache Iceberg.
Основная цель:
Протестировать и оценить работоспособность, производительность и потенциальную применимость коннектора Kafka Connect Iceberg Sink для организации непрерывной доставки данных из корпоративной шины данных Apache Kafka в таблицы формата Apache Iceberg. Это включает в себя настройку передачи данных, проверку их корректности в целевом хранилище и оценку стабильности работы решения.

3. Используемый инструментарий и среда тестирования

Для оперативного развертывания необходимой тестовой инфраструктуры был применен разработанный Исполнителем инструмент Data Platform Deployer (дальше dpd). С помощью dpd была сгенерирована и развернута тестовая среда, включающая следующие ключевые компоненты:

- Apache Kafka: Кластер из нескольких брокеров для имитации производственной шины данных.
- Kafka Connect: Распределенная платформа для запуска коннекторов, включая тестируемый Iceberg Sink Connector.

- S3-совместимое хранилище (Minio): Для хранения данных таблиц Apache Iceberg, так как Iceberg часто использует объектные хранилища в качестве основного бэкенда.
- Инструменты мониторинга (AKHQ): Для отслеживания состояния топиков Kafka и коннекторов.
- Источник данных (например, PostgreSQL с Debezium): Для генерации тестового потока данных в Kafka, имитирующего бизнес-события.

Использование инструмента dpd позволило значительно сократить время на подготовку и конфигурацию тестового стенда, обеспечив стандартизированную и воспроизводимую среду.

4. Ход выполнения работ

Генерация и развертывание инфраструктуры:

Был подготовлен конфигурационный файл для dpd, описывающий необходимые компоненты (Kafka, Kafka Connect, Minio).

С помощью команды `dpd generate --config <config_file.yaml>` был сгенерирован `docker-compose.yml` и сопутствующие файлы. Платформа была развернута командой `docker compose up -d`.

Настройка Kafka Connect Iceberg Sink Connector:

Тестируемый Iceberg Sink Connector (от Tabular) был загружен на узлы Kafka Connect.

Была создана конфигурация для Iceberg Sink Connector. В конфигурации были указаны:

- Адреса Kafka-брокеров.
- Имена топиков Kafka для чтения данных.
- Параметры подключения к Iceberg-каталогу (например, REST-каталог или Hive Metastore, если используется).
- Параметры S3-хранилища (Minio): эндпоинт, ключи доступа, бакет для хранения данных Iceberg.
- Схема данных и параметры целевых таблиц Iceberg.
- Политики коммита данных в Iceberg по времени

Конфигурация была отправлена в Kafka Connect REST API для запуска коннектора.

Генерация тестового потока данных:

В один или несколько Kafka-топиков, указанных в конфигурации коннектора, подавался тестовый поток сообщений в формате JSON, имитирующий реальные бизнес-события.

Мониторинг и проверка:

Отслеживалось состояние Iceberg Sink Connector через Kafka Connect REST API и/или AKHQ (статус, наличие ошибок, лаг).

Проверялось создание и наполнение таблиц Apache Iceberg в S3-хранилище (Minio) в соответствии с ожидаемой структурой и форматом (например, Parquet или ORC файлы данных, файлы метаданных Iceberg).

Выполнялись тестовые запросы к данным в таблицах Iceberg с использованием совместимого Spark SQL, чтобы убедиться в корректности и доступности записанных данных.

5. Результаты тестирования

Развертывание инфраструктуры: Тестовая среда с использованием dpd была успешно и быстро развернута, все компоненты (Kafka, Kafka Connect, Minio) функционировали корректно.

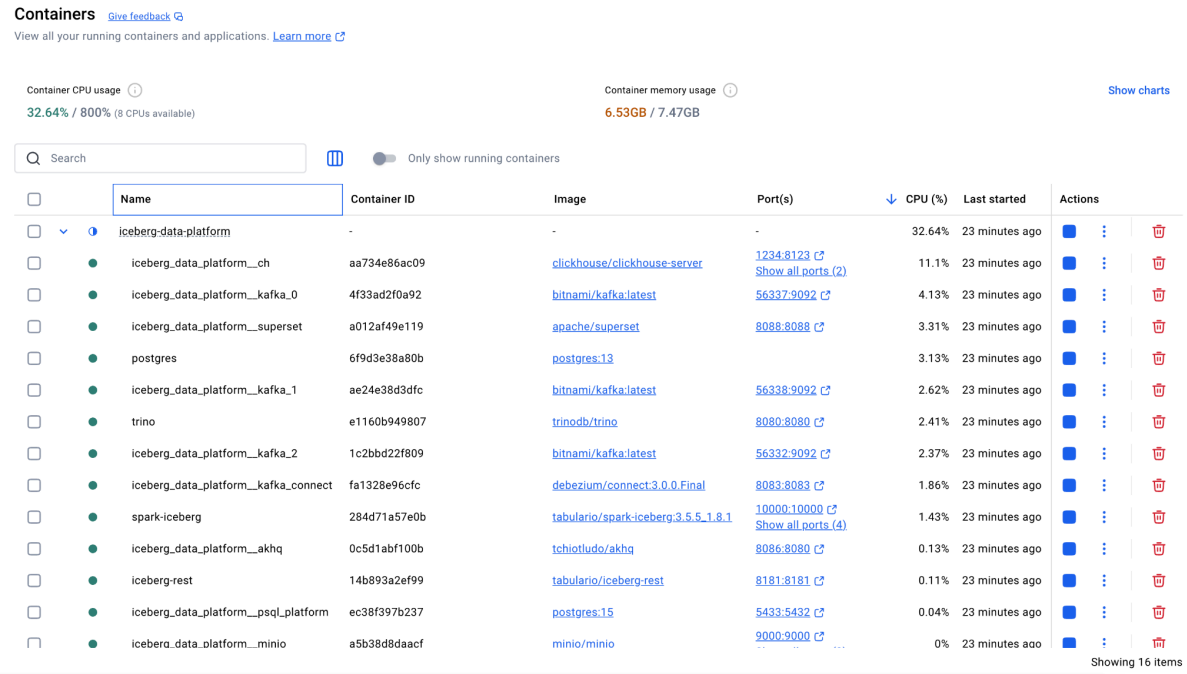


Рис. 1 Развернутая инфраструктура в Docker Desktop

Работа коннектора: Kafka Connect Iceberg Sink Connector был успешно запущен и настроен. Коннектор стабильно потреблял данные из указанных Kafka-топиков.

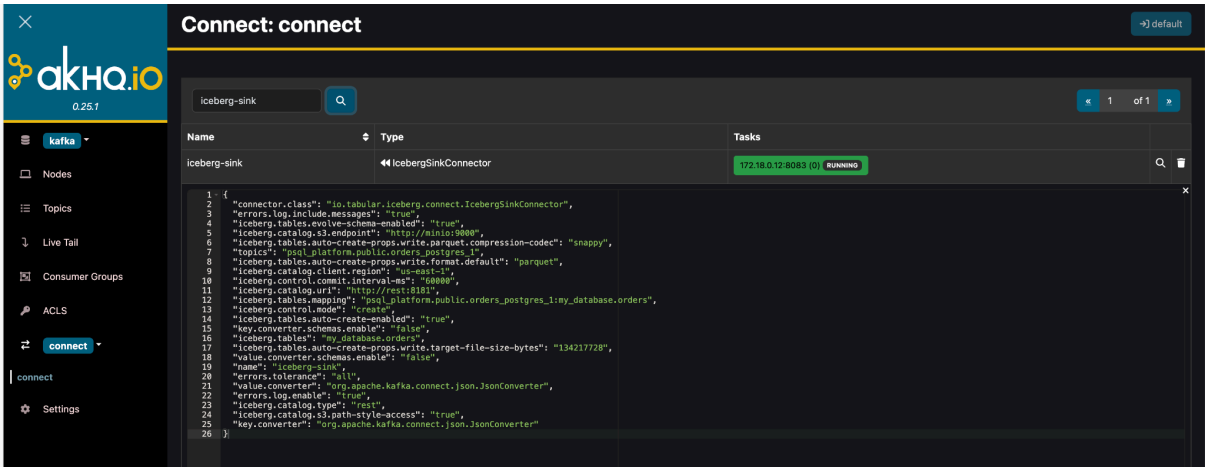


Рис. 2 Конфигурация коннектора IcebergSinkConnector в AKHQ (Kafka UI)

Передача данных: Данные из Kafka успешно передавались и записывались в таблицы Apache Iceberg. Наблюдалась корректная сериализация/десериализация сообщений и их преобразование в формат, пригодный для Iceberg.
Структура в Iceberg: В S3-хранилище (Minio) корректно создавались директории и файлы, соответствующие структуре таблиц Iceberg (файлы данных, манифесты, файлы метаданных).

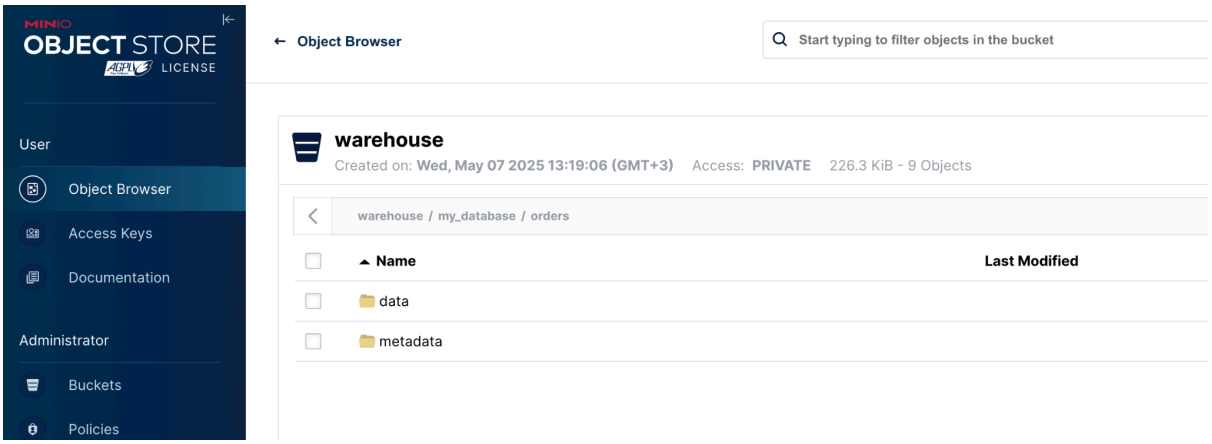


Рис. 3 Структура директории таблицы orders в Minio UI

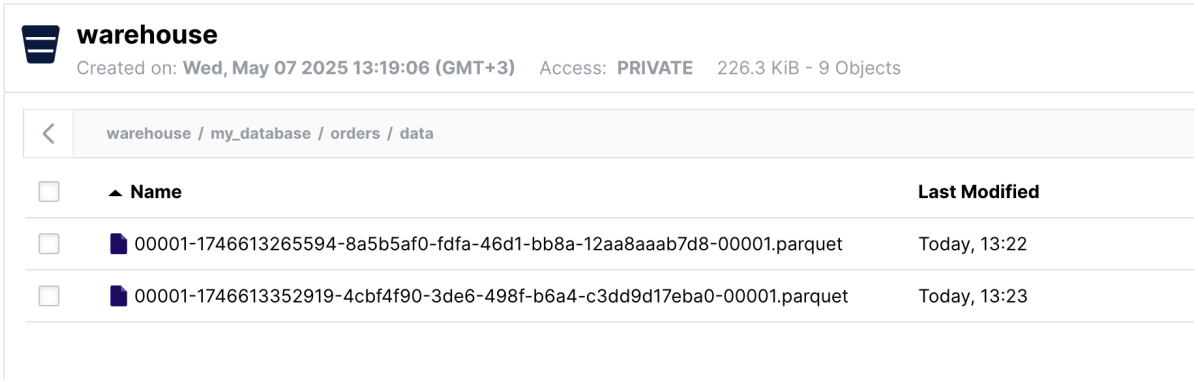


Рис. 4 Содержимое директории data в Minio UI

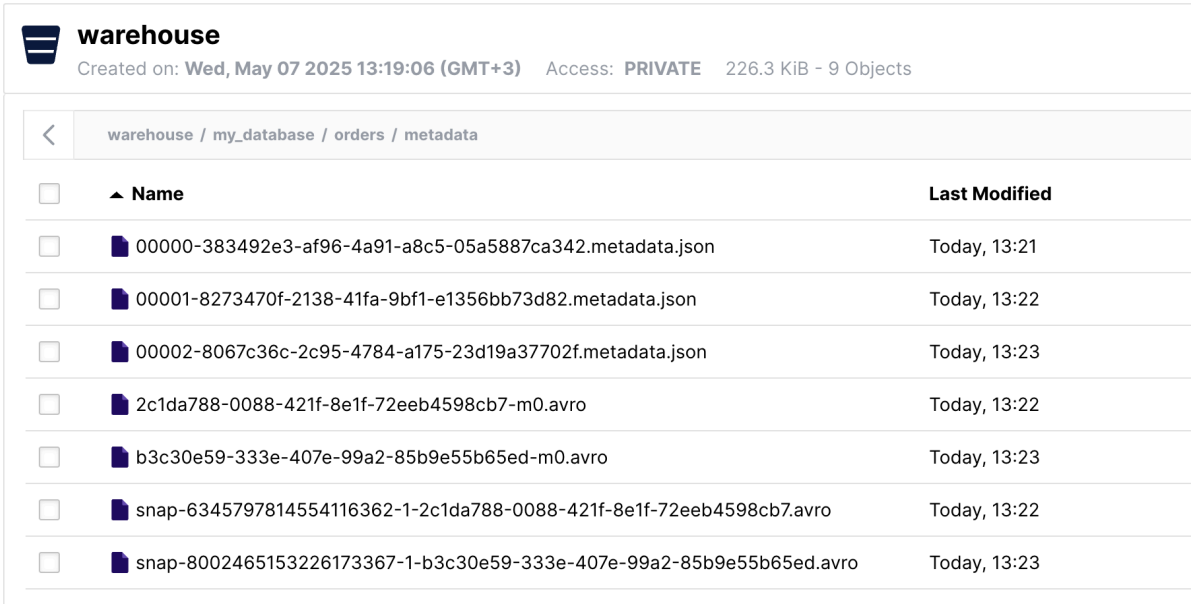
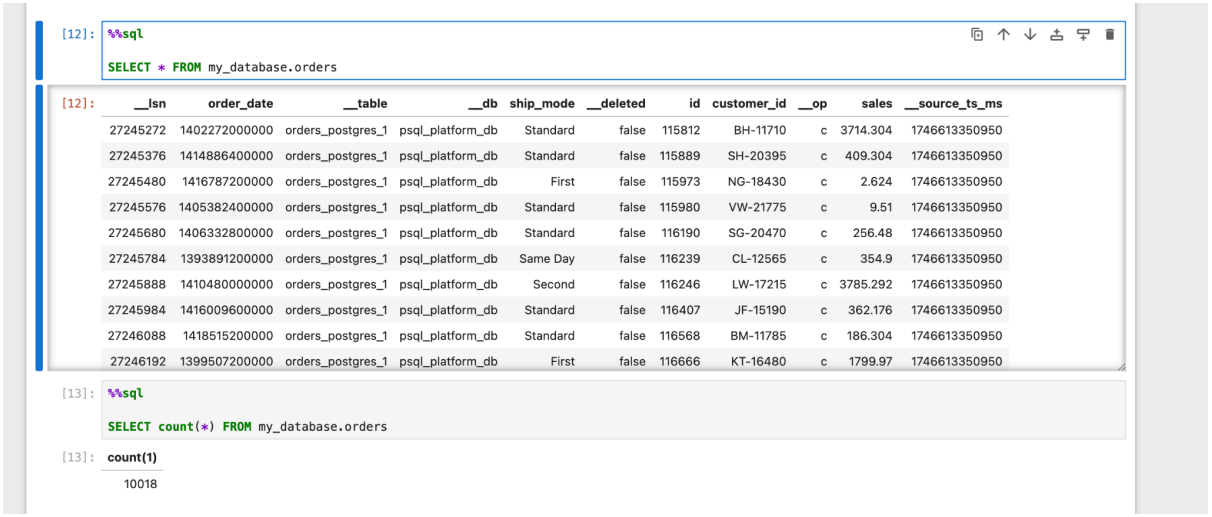


Рис. 5 Содержимое директории metadata в Minio UI

Целостность и доступность данных: Выборочная проверка данных, записанных в Iceberg, показала их соответствие исходным данным в Kafka. Данные были успешно прочитаны с помощью аналитических инструментов, совместимых с Iceberg.



```
[12]: %%sql
SELECT * FROM my_database.orders
```

[12]:	__lsn	order_date	__table	__db	ship_mode	__deleted	id	customer_id	__op	sales	__source_ts_ms
	27245272	1402272000000	orders_postgres_1	psql_platform_db	Standard	false	115812	BH-11710	c	3714.304	1746613350950
	27245376	1414886400000	orders_postgres_1	psql_platform_db	Standard	false	115889	SH-20395	c	409.304	1746613350950
	27245480	1416787200000	orders_postgres_1	psql_platform_db	First	false	115973	NG-18430	c	2.624	1746613350950
	27245576	1405382400000	orders_postgres_1	psql_platform_db	Standard	false	115980	VW-21775	c	9.51	1746613350950
	27245680	1406332800000	orders_postgres_1	psql_platform_db	Standard	false	116190	SG-20470	c	256.48	1746613350950
	27245784	1393891200000	orders_postgres_1	psql_platform_db	Same Day	false	116239	CL-12565	c	354.9	1746613350950
	27245888	1410480000000	orders_postgres_1	psql_platform_db	Second	false	116246	LW-17215	c	3785.292	1746613350950
	27245984	1416009600000	orders_postgres_1	psql_platform_db	Standard	false	116407	JF-15190	c	362.176	1746613350950
	27246088	1418515200000	orders_postgres_1	psql_platform_db	Standard	false	116568	BM-11785	c	186.304	1746613350950
	27246192	1399507200000	orders_postgres_1	psql_platform_db	First	false	116666	KT-16480	c	1799.97	1746613350950

```
[13]: %%sql
SELECT count(*) FROM my_database.orders
```

```
[13]: count(1)
      10018
```

Рис. 6 Запросы в Iceberg таблицу в SparkSQL

Производительность (качественная оценка): При тестовой нагрузке коннектор демонстрировал приемлемую производительность, лаг потребления из Kafka оставался в допустимых пределах. Учитывая параметр `iceberg.control.commit.interval-ms` лаг был ненулевой в пределах 3 минут, при лаге 60000 сообщений.

Замечания и наблюдения:

- при использовании `catalog.type = hive` возникала ошибка

```
lang.IllegalArgumentException: Cannot initialize Catalog
mentation org.apache.iceberg.hive.HiveCatalog: Cannot find
ructor for interface org.apache.iceberg.catalog.Catalog
ng org.apache.iceberg.hive.HiveCatalog
.lang.ClassNotFoundException:
pache.iceberg.hive.HiveCatalog]
g.apache.iceberg.CatalogUtil.loadCatalog(CatalogUtil.java:2
```

Что говорит о конфликтах при работе с `hive`, поэтому текущая реализацию использует `rest`

- при попытки посмотреть данные Iceberg из под Trino возникала ошибка

```
rino. spi. TrinoException: Error processing metadata for ta
tabase. orders java. lang. IllegalArgumentException: No fac
ocation: s3:// warehouse/ my_database/ orders/ metadata/
80024651532261733 67-1-b3c30e59-333e-407e-99a2-85b9e55b65ed.
```

Чтобы обойти эту ошибку нужно изменить настройки подключения к Iceberg со стороны Trino, конфигурация `trino/iceberg.properties`, реализация с SparkSQL не потребовала точной настройки.

6. Заключение и выводы

По результатам проведенного внедрения и тестирования можно сделать следующие выводы:

Технология Kafka Connect Iceberg Sink Connector является работоспособным и перспективным решением для организации потоковой загрузки данных из Apache Kafka в хранилище данных Apache Iceberg.

Протестированная связка компонентов (Kafka -> Kafka Connect -> Iceberg Sink -> Iceberg (на S3)) продемонстрировала стабильную работу и корректную передачу данных в рамках тестовой среды.

Использование разработанного инструмента dpd существенно упростило и ускорило процесс подготовки и развертывания необходимой инфраструктуры для тестирования, что подтверждает его практическую ценность для решения подобных инженерных задач.

Полученные результаты позволяют рекомендовать дальнейшее, более углубленное исследование и пилотирование Kafka Connect Iceberg Sink Connector для возможного применения в продуктивных системах ПАО «Магнит» с целью построения современных озер данных на базе Apache Iceberg.

Настоящий протокол подтверждает факт выполнения поставленной рабочей задачи и успешную апробацию разработанного инструмента dpd в условиях, приближенных к реальным задачам компании.

Подписи:

Исполнитель:

_____ / (Ф.И.О.) /
(подпись)
Ответственное лицо от ПАО «Магнит» (Руководитель):

_____ / (Ф.И.О., Должность) /
(подпись)

«__» _____ 2025 г.
М.П. (если применимо)