детский мир

дм тех

детский мир

дм тех

dbt в деле: реальные кейсы и лайфхаки

Александров Антон

Содержание

1. О докладчике и Группе компаний Детский мир
2. Начало
    1. BigData
    2. Задачи
3. Первые подходы к dbt
4. Организация CI/CD
5. Подготовка Аналитиков
6. Деплой в Airflow
7. Новые вызовы
8. Итоги

01 О докладчике

Александров Антон

1. Обучение
МИИТ Прикладная математика и Информатика
МФТИ Прикладная математика и физика

2. Предыдущие проекты

Beeline - Performance Management платформа
200 GB raw data /day
Работа с MapReduce API

Luxoft - Озеро данных Почты России/ Отчеты для команды Логистики
3 PB Hadoop Cluster size

SeverGroup - собственная платформа ClickStream аналитики
Использование ClickHouse как основного хранилища

Ростелеком – Создание клиентского профиля
1.5 TB/day
Uid matching
Внедрение ML моделей в Production
Решение проблем производительности

3. Суммарный опыт работы с Big Data tools **9 лет**

4. В 2022 команда Data Office была 6 человек
Текущий размер команды **25** человек

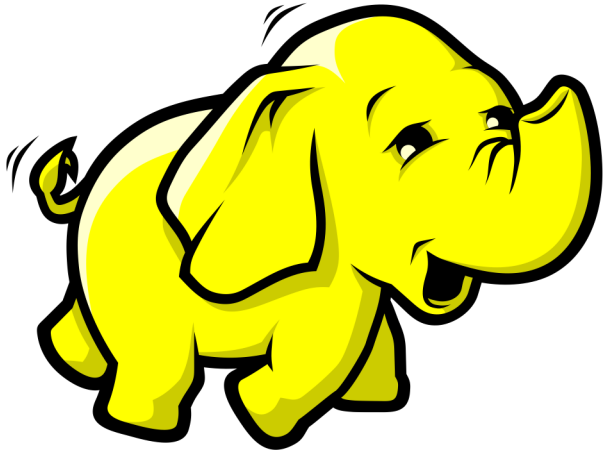
Группа компаний «Детский мир» сегодня

- 3 страны присутствия
- более 1300 магазинов
- более 440 городов
- более 20 тысяч сотрудников
- 6 распределительных центров
- более 900 тысяч SKU онлайн-ассортимента
- более 600 млн онлайн-посещений ежегодно
- более 250 млн посещений магазинов в год
- 12+ млн пользователей мобильного приложения
- более 42 тысяч пунктов выдачи
- более 7000 поставщиков маркетплейса

02 Начало BigData в DM

Детский мир BigData 2022

Источники данных



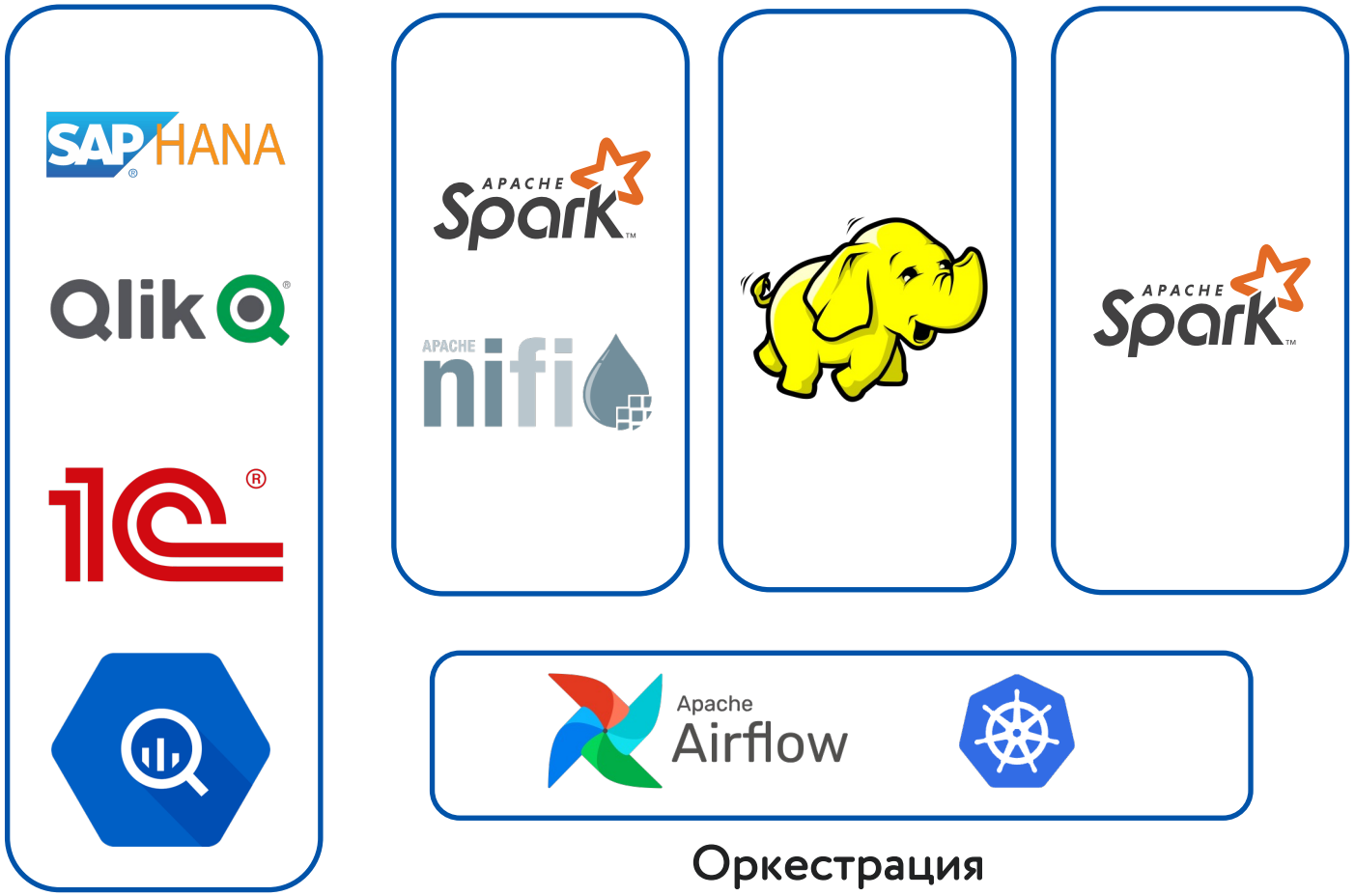
Omni Dashboard / CEO

Источники данных

Загрузка

Хранение

Манипуляции с данными



Оркестрация

омни-дашборд

08 февраля 2023

GMV

RUB, млн	Бюдж.	YoY	
[illegible]	98 %	6 %	Д
[illegible]	105 %	12 %	М
[illegible]	102 %	11 %	Г

Фронт-маржа

Ф.-маржа	Бюдж.	Бюдж., RUB	YoY	
18,5 %	(1,2 пп)	92 %	8,8 пп	Д
19,7 %	(0,2 пп)	103 %	8,3 пп	М
19,1 %	(0,6 пп)	99 %	6,1 пп	Г

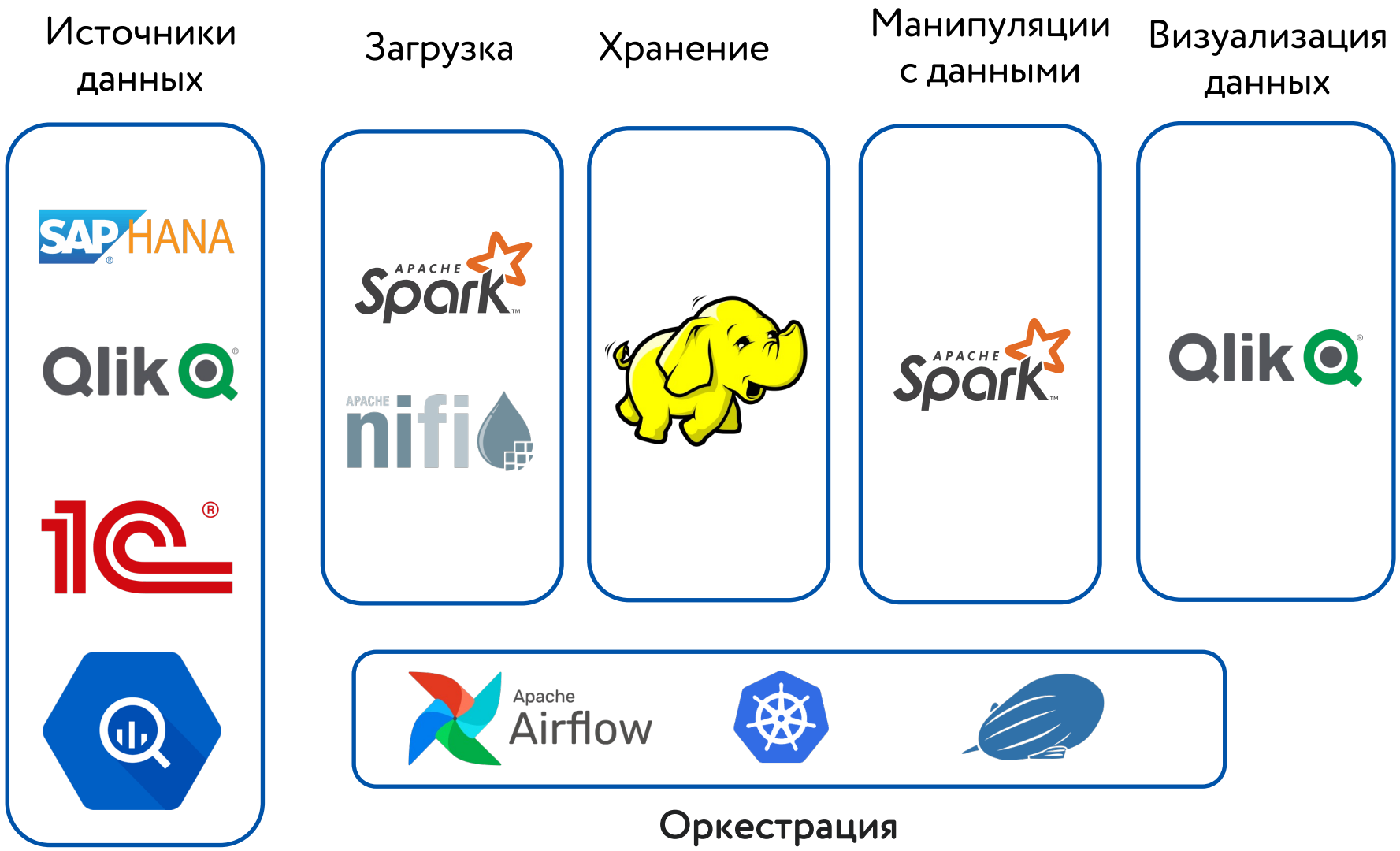
GMV онлайн

Доля от общ. GMV	RUB, млн	Бюдж.	YoY	
34,3 %	[illegible]	88 %	(10 %)	Д
31,8 %	[illegible]	93 %	(8 %)	М
28,7 %	[illegible]	101 %	(4 %)	Г

Фронт-маржа онлайн

Ф.-маржа	Бюдж.	Бюдж., RUB	YoY	
7,5 %	(3,7 пп)	67 %	7,9 пп	Д
7,4 %	(3,8 пп)	70 %	8,5 пп	М
7,7 %	(2,8 пп)	83 %	6,5 пп	Г

Omni Dashboard / C Level Reporting / Version 1



Omni Dashboard / C Level Reporting / Version 1

Переработка SQL-запроса для загрузки в Qlik Sence.

Описание

Редактировать описание

Вложения 7

15 нояб. 2022, 12:11 PM

10 окт. 2022, 07:26

Связанные задачи

blocks

Загрузка новых SQL-запросов по замене показателя

Описание

DOD

1. Обновить справочник
2. Обновить sql
   a.
   b.

Вложения 6

Создать витрину по продажам онлайн

Описание

Прошу создать новую витрину по продажам онлайн в схеме dashboard. Код формирования витрины во вложении.

Таблицу можно назвать

Вложения 1

Создание витрины с данными по товарному запас

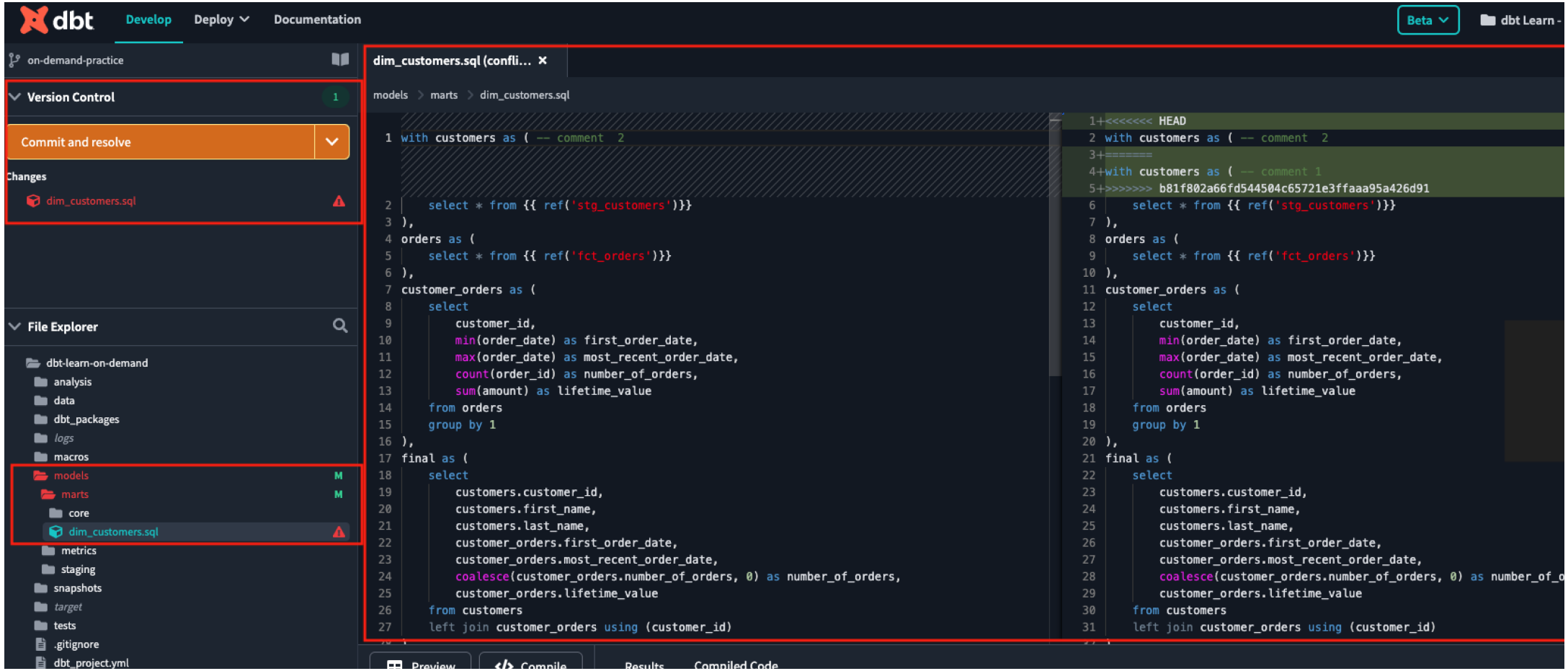
Описание

Создайте, пожалуйста, новую витрину в Hadoop (схема Dashboard). Код во вложении. Назвать таблицу можно

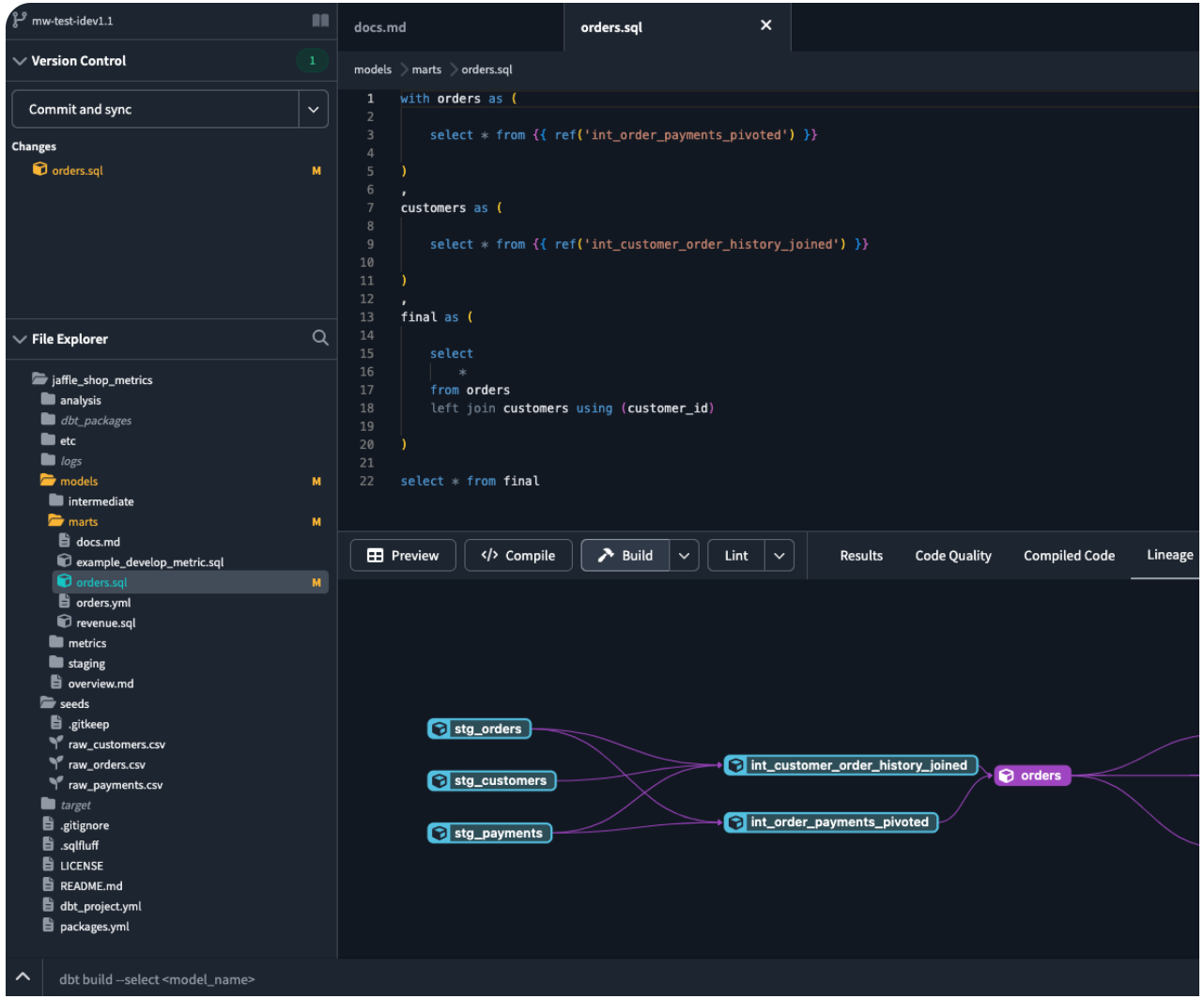
Вложения 1

03 Первые подходы к dbt

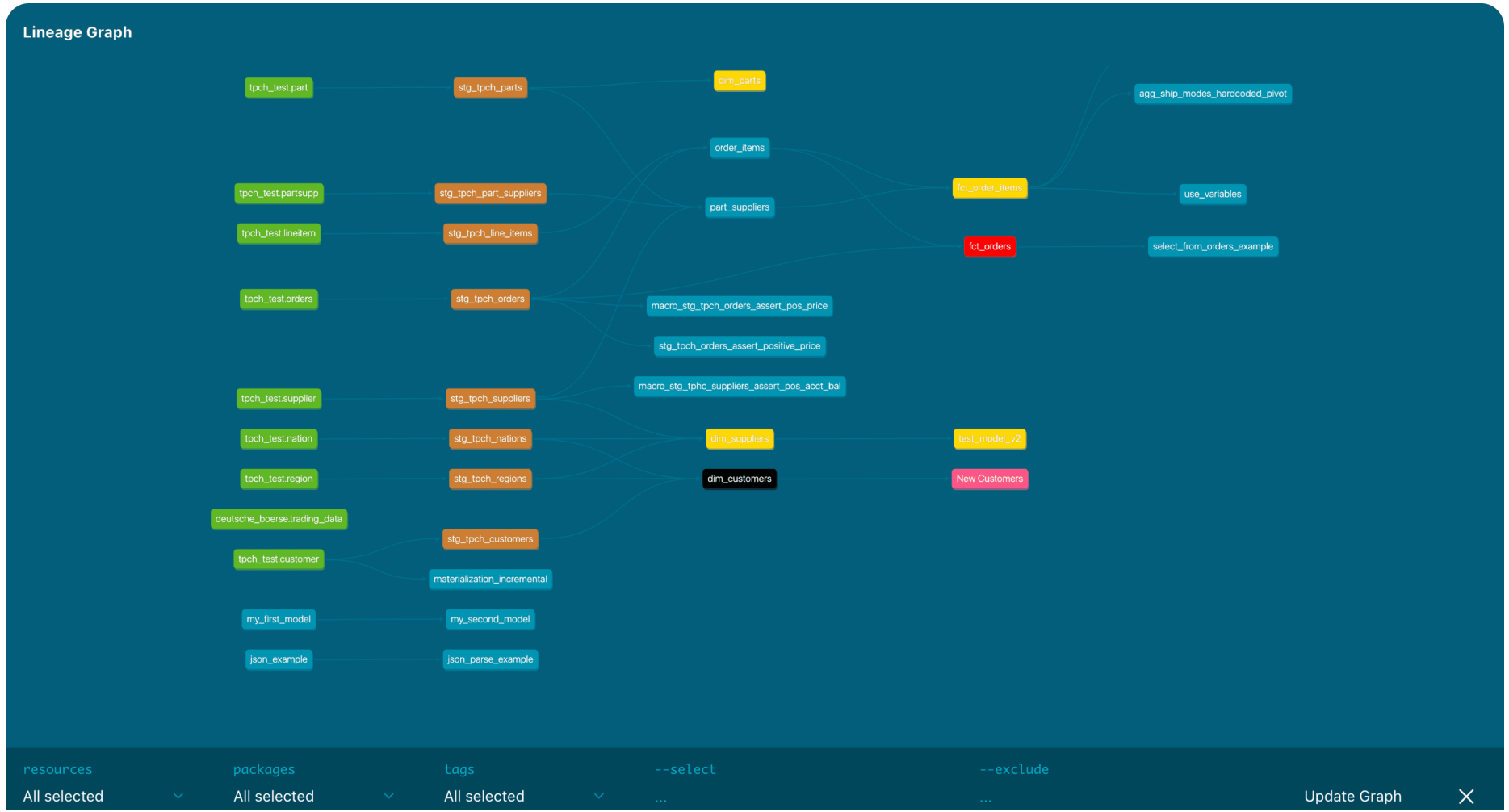
Коротко о DBT / Git ops хранение



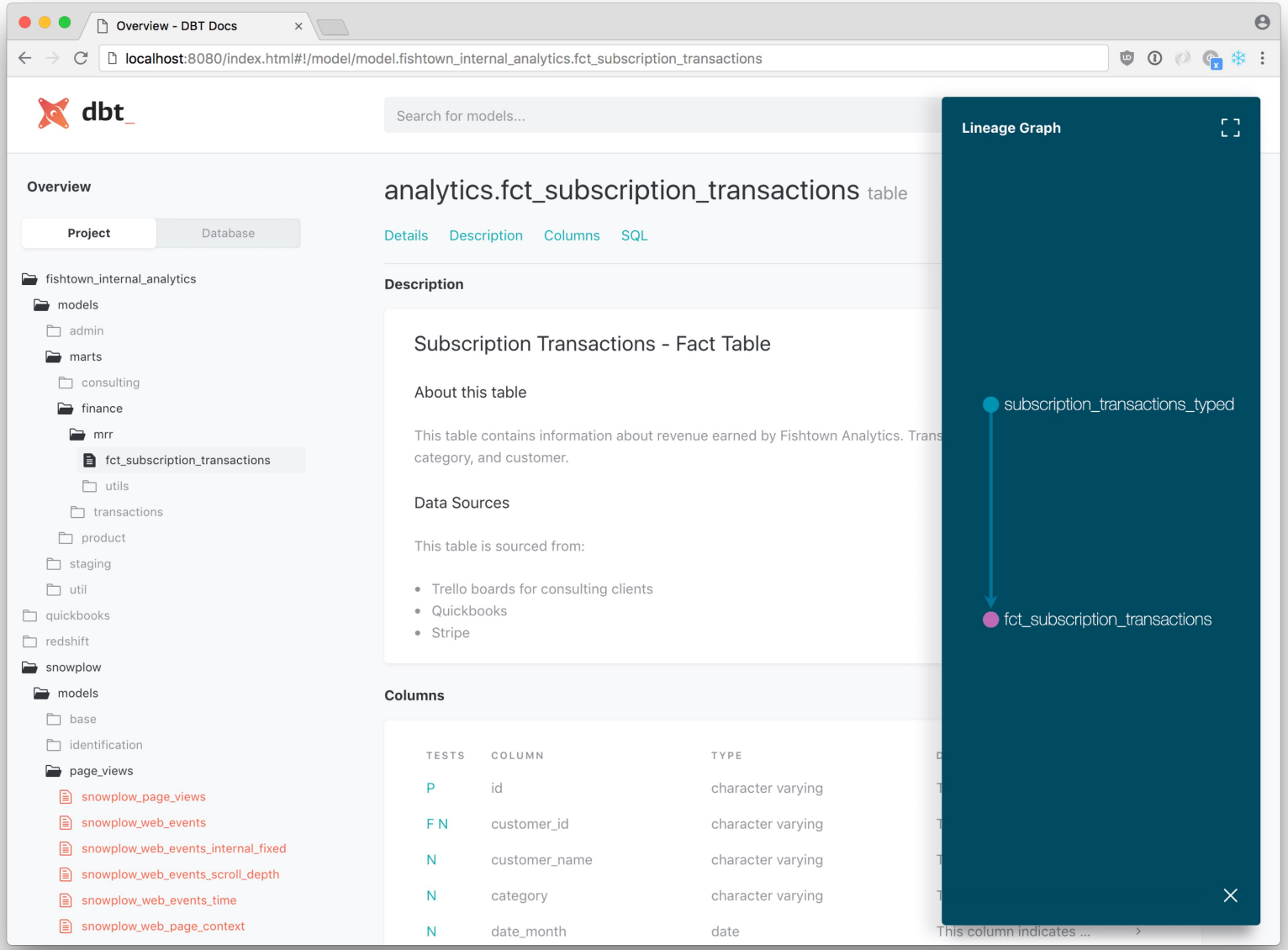
Коротко о DBT / Построение зависимостей



Коротко о DBT / Data lineage



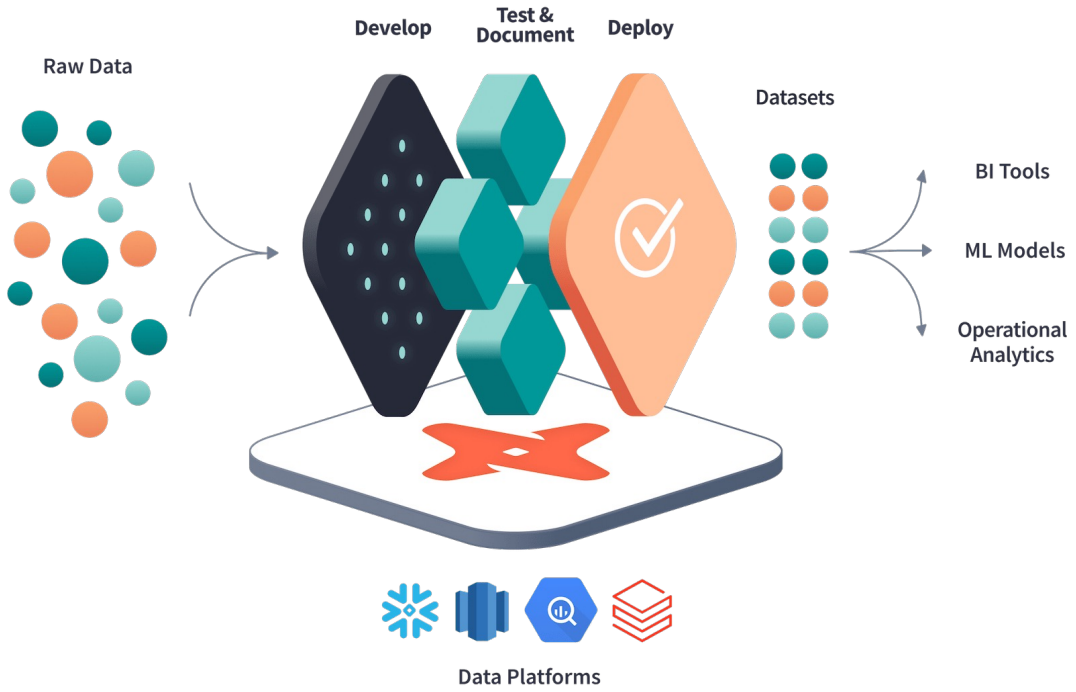
Коротко о DBT / Документация



DBT / Все вместе

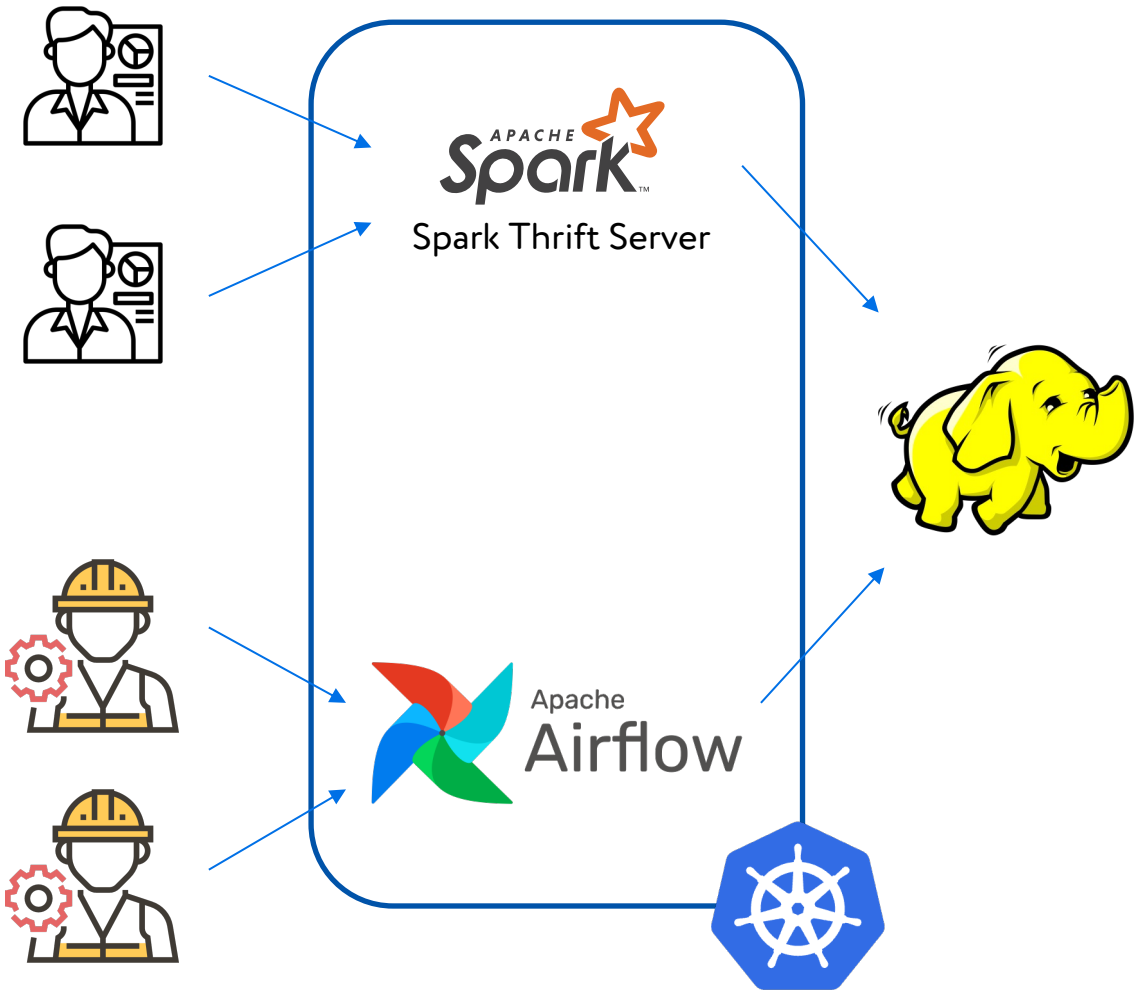
- SQL без порядка это больно
- Добиться порядка в SQL сложно

- Витрины данных по методологии с GitOps
- Порядок из коробки
- Значительное ускорение Time To Market
- Поиск зависимостей в витринах данных
- Довольные Аналитики и Инженеры



DBT / Про стек

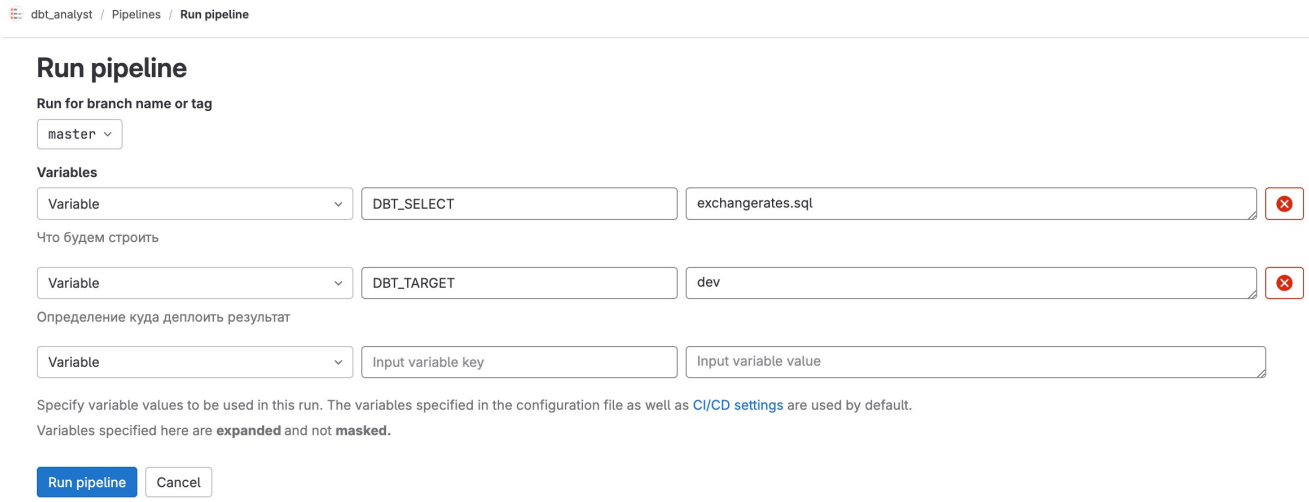
- Наш стек
  - Hadoop
  - SparkThriftServer
  - Airflow in K8S

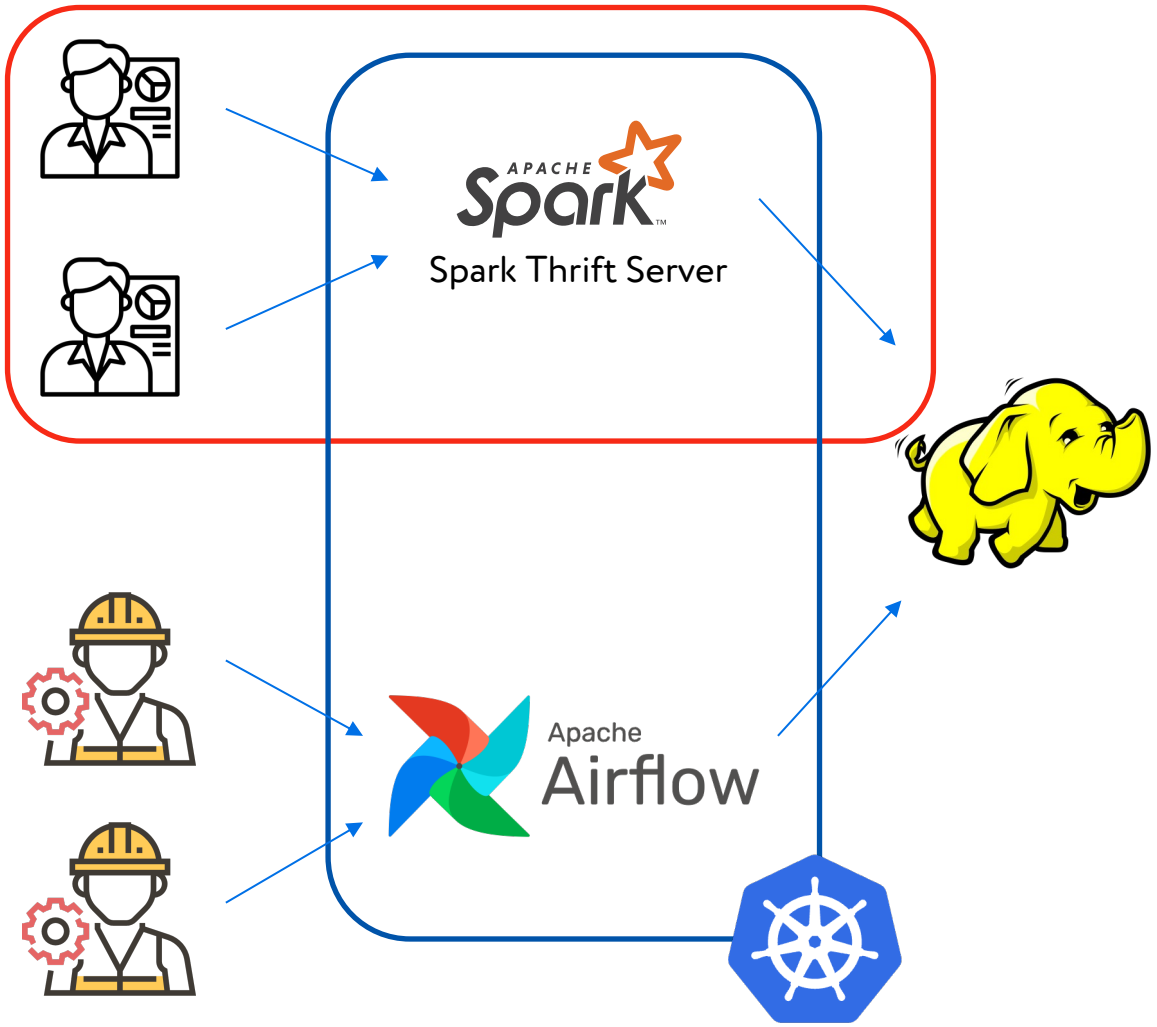


DBT / работа с Hadoop для аналитиков

- Есть адаптер dbt-spark
- Все хорошо заработал сразу
- Из винды не запускаются построения
- Для этого сделали в GitLab пайплайн который

запускает построение определенной модели

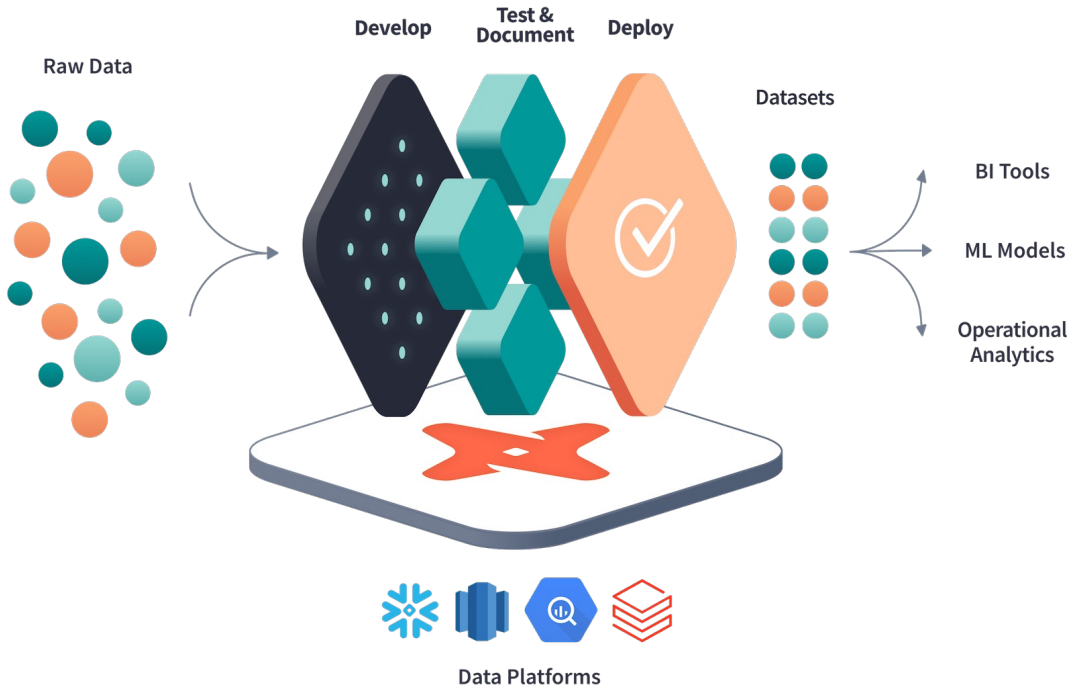




DBT / Генерация документации

Генерируем доку
Складываем в GitLab Pages

```
pages:
  stage: deploy
  cache:
    - key: target
      paths:
        - public
  variables:
    DBT_PROFILES_DIR: "${CI_PROJECT_DIR}/omni_1"
  script:
    - cd omni_1
    - dbt debug -t prod
    - dbt deps -t prod
    - dbt docs generate -t prod
    - mkdir -p ${CI_PROJECT_DIR}/public/$CURRENT_CONTENT_PATH
    - cp -r ${CI_PROJECT_DIR}/omni_1/target/*
${CI_PROJECT_DIR}/public/$CURRENT_CONTENT_PATH
  artifacts:
    paths:
      - public
    expire_in: 4 weeks
```



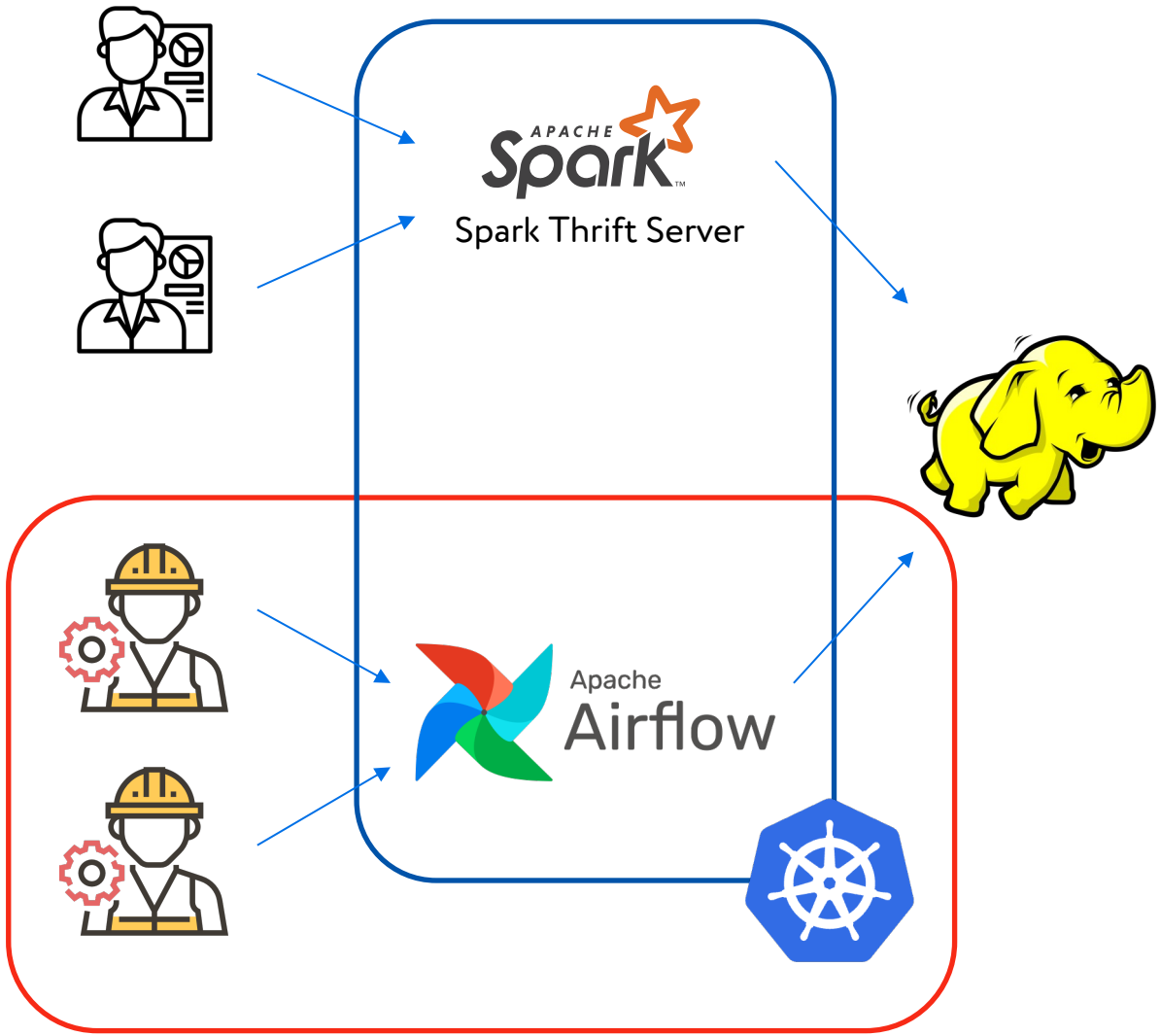
DBT / Интеграция в airflow

```python
def load_manifest():
    local_filepath = f"{DBT_PROJECT_DIR}/target/manifest.json"
    with open(local_filepath) as f:
        data = json.load(f)
    return data
```

- На 2022 был только cosmos в зачаточном состоянии
- На 2024 есть
  - Astronomer cosmos
  - Toloka / dbt-af

DBT / работа с Hadoop для инженеров

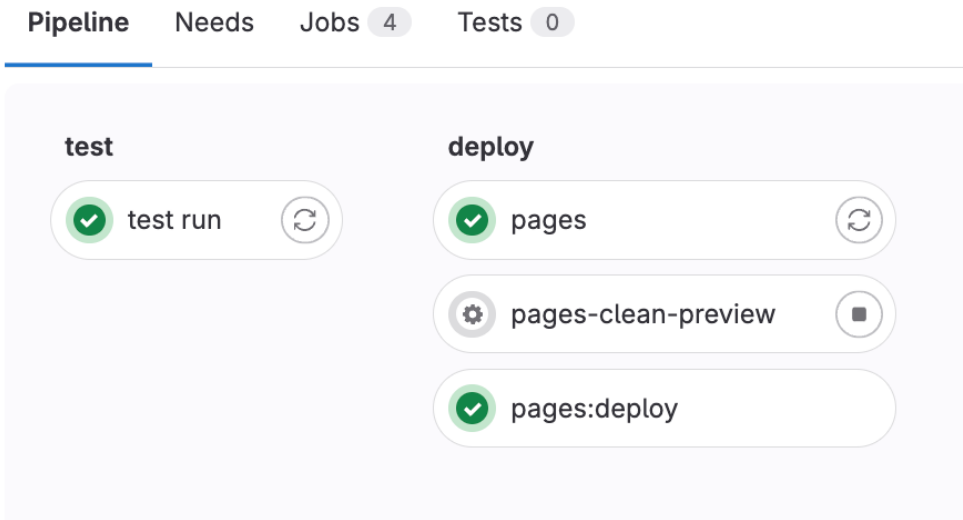
- не подходит sparkSqlOperator
  - Из за запуска в кубере сеть виртуализирована, executor не могут общаться с драйвером
- Dbt создает артефакты
  - manifest.json
    - построение графа
    - содержит мету о тегах
  - Скопилированный sql
    - Запуск с помощью спарксабмит




```
val source = Source.fromFile(queryPath.mkString.split("/").last, "UTF-8")
val query = try source.mkString finally source.close()

spark
  .sql(query)
  .repartition(numPartitions.toInt)
  .write
  .mode(SaveMode.Overwrite)
  .format("parquet")
  .saveAsTable(resultTable)
```

04 Организация CI/CD для dbt

CI/CD для dbt repo




```
test run:
  stage: test
  image: dbt:rel_1.8.4_u1
  variables:
    DBT_PROFILES_DIR: "${CI_PROJECT_DIR}/omni_1"
  script:
    - python validation/validation.py --models_directory
${CI_PROJECT_DIR}/omni_1/models
    - cd omni_1
    - dbt debug -t dev_view --project-dir ../ci_cd
    - dbt deps -t dev_view --project-dir ../ci_cd
    - dbt compile -t dev_view --project-dir ../ci_cd
    - dbt -x ls -t dev_view --project-dir ../ci_cd
    - dbt run -t dev_view --project-dir ../ci_cd
    - dbt test -t dev_view --project-dir ../ci_cd
```

- Нужно что аналитики сразу знали фидбек на свои изменения
- Организовано построение вьюх
  - понимаем, что витрины валидны технически и мы можем их ставить на исполнение
- В шаг CI/CD добавляем:
  - Защиту от прямого использования таблиц
  - Если не используется {{ref }} {{source}}, падаем

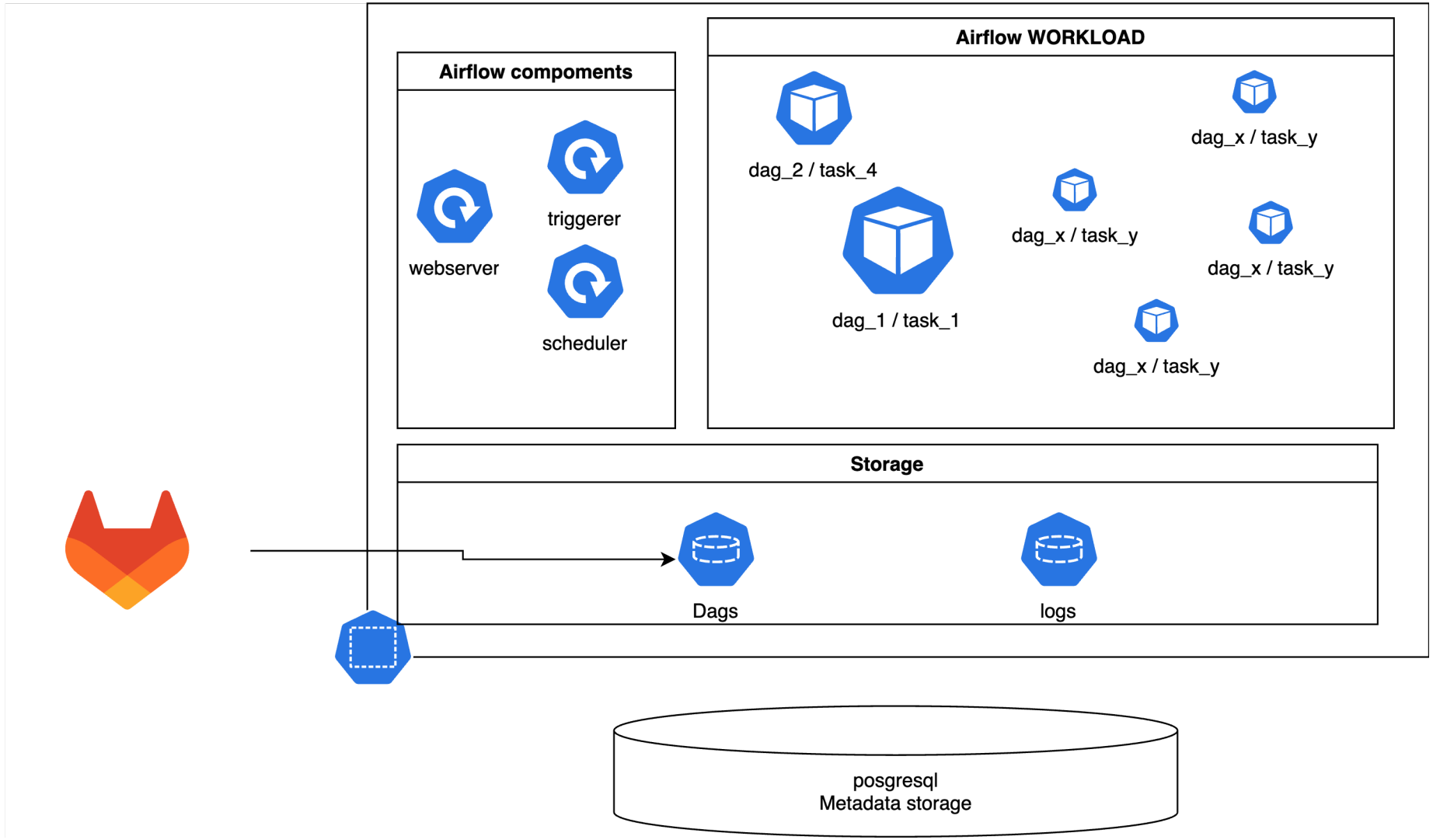
05 Подготовка Аналитиков

Рассказываем о новом процессе

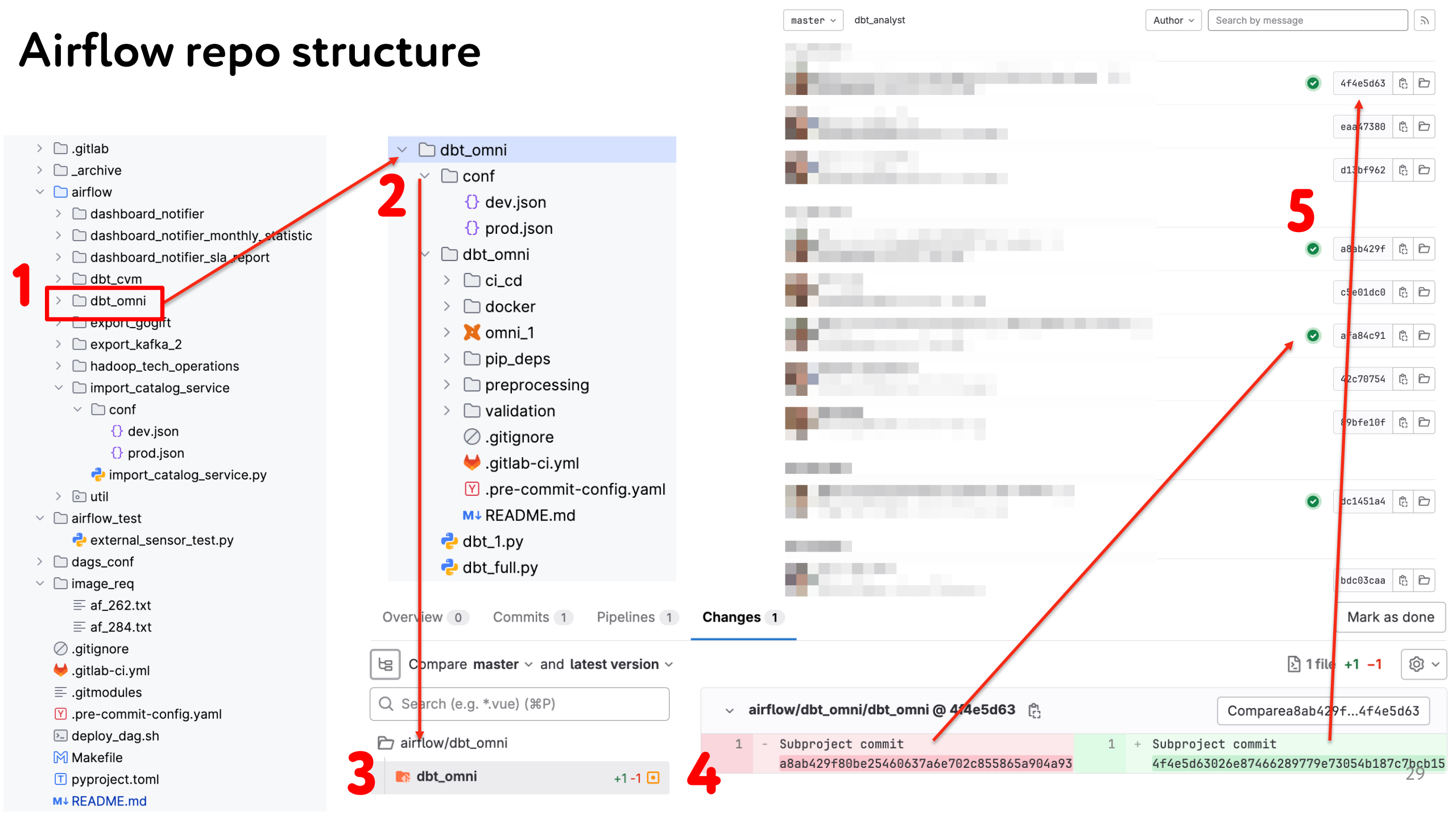
- Вспоминаем, для кого мы все это делали
- Заранее проверяем хотят ли аналитики этих оптимизаций
- Презентуем изменения до готовности
- Обучение git в формате workshop
- Запирались в переговорке и учились делать коммиты в ветки
- Настройки локальных окружений
- Настраиваем одному, чтобы он рассказал остальным

06 Деплой в Airflow

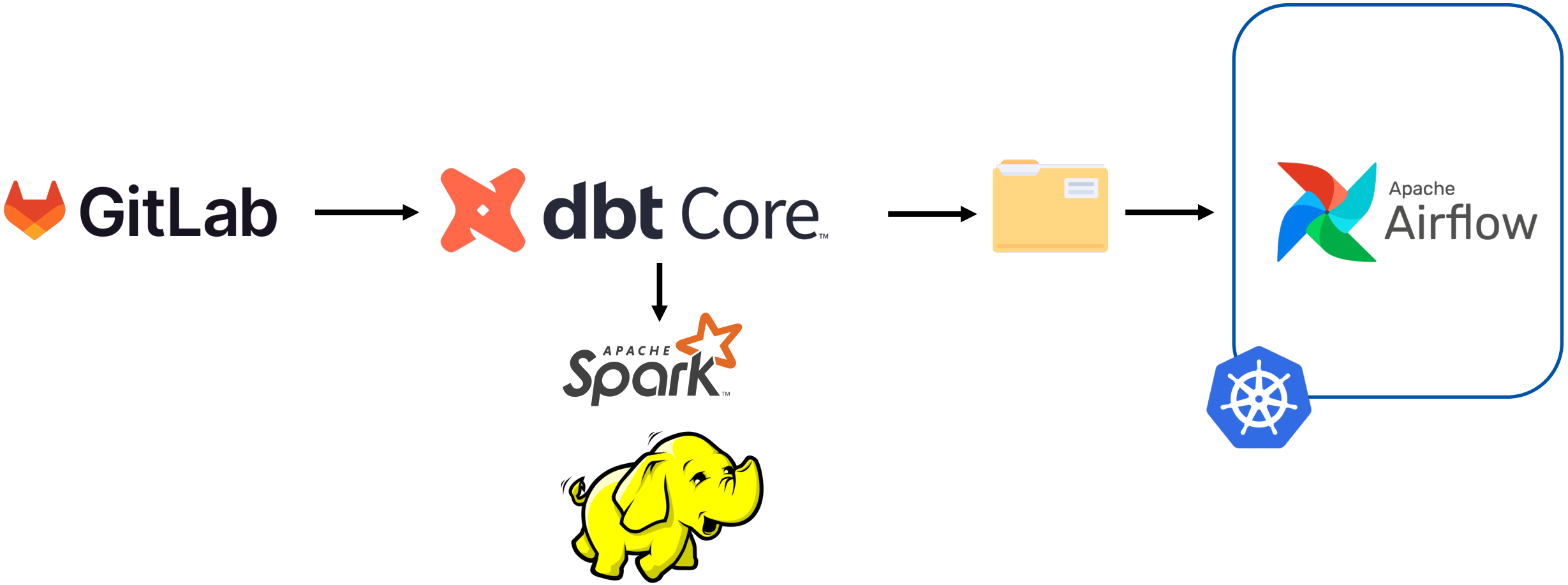
Airflow k8s architecture



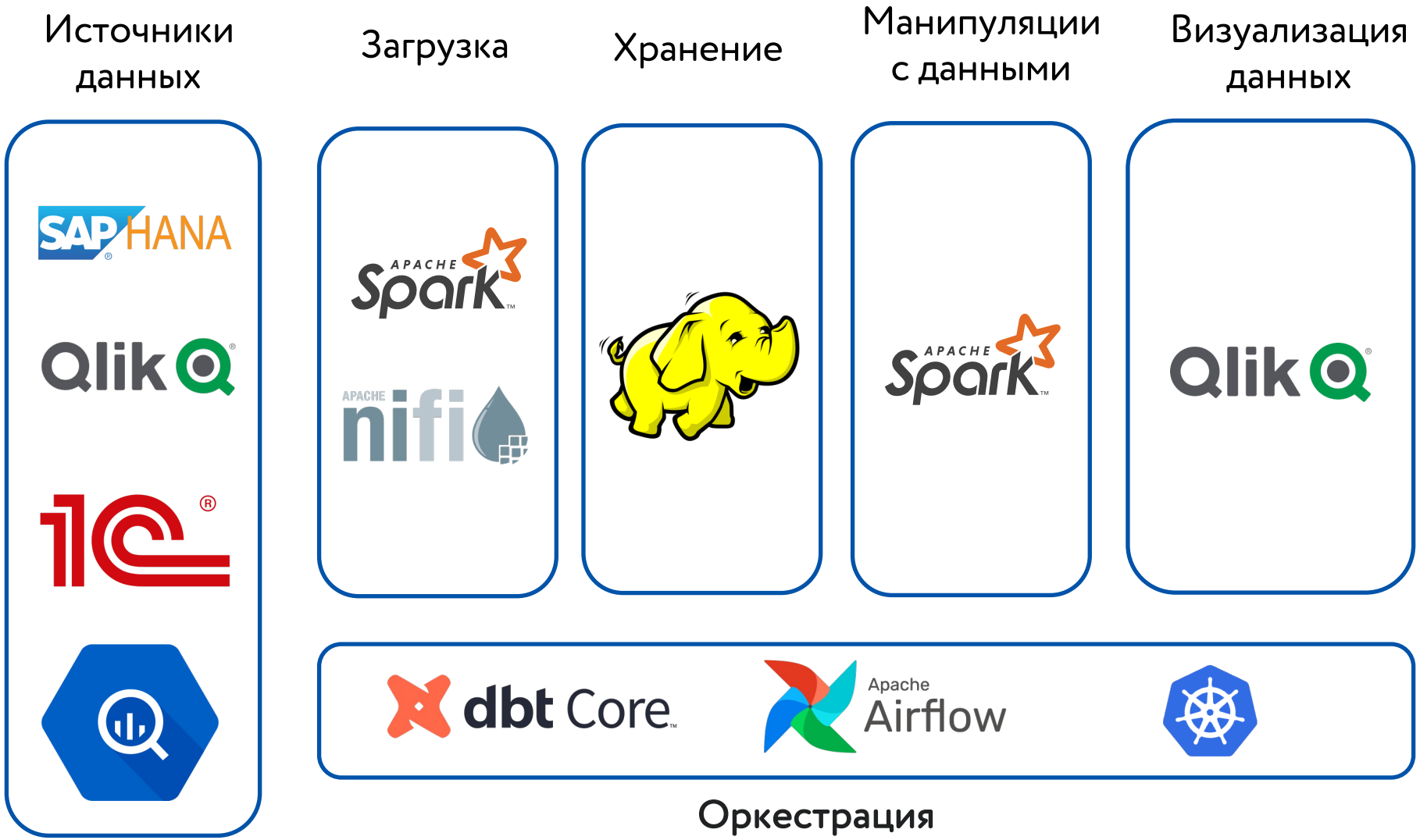
Airflow repo structure



Subrepository model



Omni Dashboard / C Level Reporting / Version 2



Сравнение процессов разработки витрин

Поток доставки задачи как изменился

Было

DA пишет 1-3 sql витрины

Отправляет их DE в задаче в jira

DE начинает проверять sql

DE разбирается, что за чем идет

DE строит view на базе витрин

DE пишет большой даг файл с нашими sql

Стало

DA пишет 1-3 sql витрины

Все витрины строятся автоматически в ci/cd,

сразу понимает что витрины валидны

DA создает Merge Requets

DE начинает смотреть, что изменилось в SQL

DE ставит апрув и мержит

DE обновляет dbt в airflow

07 Новые вызовы

Разделение на легкие/тяжелые витрины

- Сделали отдельные пулы
- Разные конфиги ресурсов
- Зашедуленные таски убивались по непонятным причинам и причем тут AIRFLOW__SCHEDULER__TASK_QUEUED_TIMEOUT

List Pool

Search

Actions

Record Count: 16

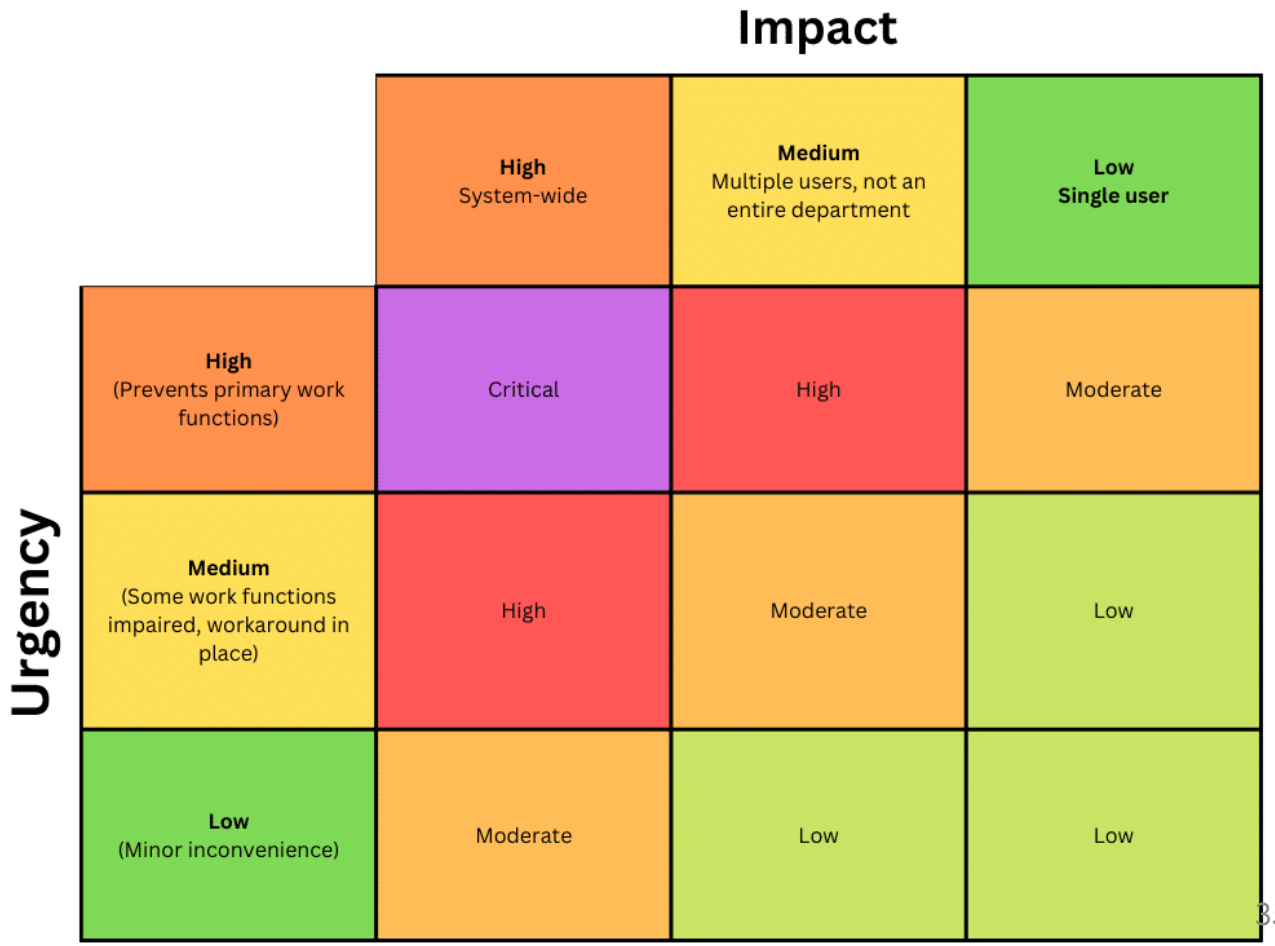
Pool	Description	Slots	Running Slots	Queued Slots	Scheduled Slots	Deferred Slots
dbt_omni	dbt_omni exec limit	2	2	0	9	0
dbt_omni_light	dbt_omni easy task exec limit	4	0	0	0	0

Витрины все нужны, но не одинаково важны

детский мир

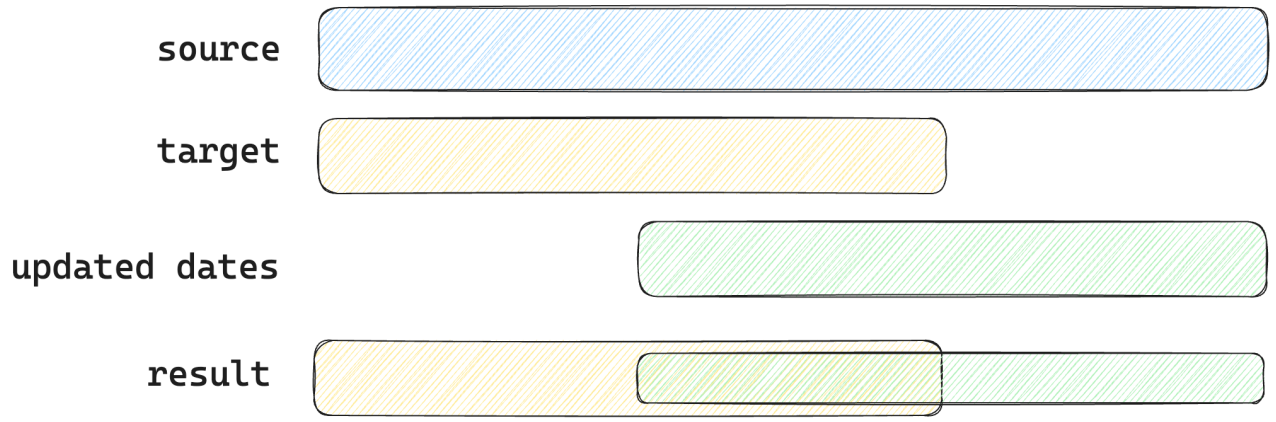
Когда готовы все сенсоры что делать в первую очередь
Сделали приоритезацию по тегам в моделях
Airflow имеет уже готовый функционал priority_weight

```
tag_weights = {
    ”aaa": 8,
    ”b": 7,
    ”c": 6,
    ”d": 5,
    ”e": 4,
    "other": 3,
    ”project_100": 2
}

dbt_task = SparkSubmitOperator(
    task_id=node,
    name=f"{DAG_ID} / {target_table} / " + "{{ ds }}",
    pool=pool,
    priority_weight=task_weight,
    weight_rule="absolute",
)
```

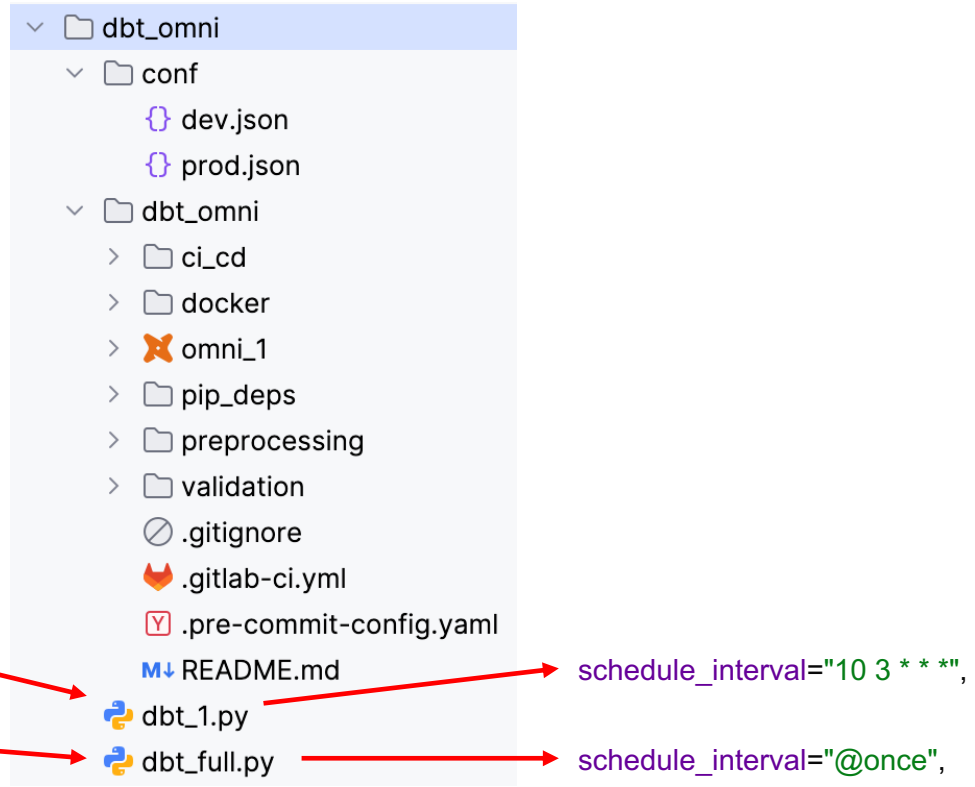


Инкрементальные витрины

```
{{ config(tags = ['aaaa'],
   spark_materialized='incremental',
   spark_partition_fields= ["calday"]) }}

SELECT a.*,

{% if var("is_incremental") | as_bool  %}
explode((sequence(current_date() - 7, current_date(), interval 1 day))) calday
      {%else%}
explode((sequence(to_date('2023-01-01'), current_date() , interval 1 day))) calday
{% endif %}

FROM (SELECT DISTINCT a,
                      b,
                      c
     from {{ source('scheme', 'table') }}) a
```



- Dbt is_incremental() не подходит, живет своей жизнью
- Сделали отдельный sparkSubmit
  - принимает колонку партиционирования
  - А так же режим выполнения full или incremental
- Создаем 2 дага
  - Основной для инкрементальных дневных расчетов
  - Дополнительный даг надо руками тригерить для полного пересчета изменённых моделей



Долгое CI/CD

Каждый коммит в airflow это проверка моделей в dbt
Проверка идет несколько минут

Написали скрипт для кеширования артефактов
Если когда данных коммит был собран то его надо собирать
Функционал уже есть в Gitlab

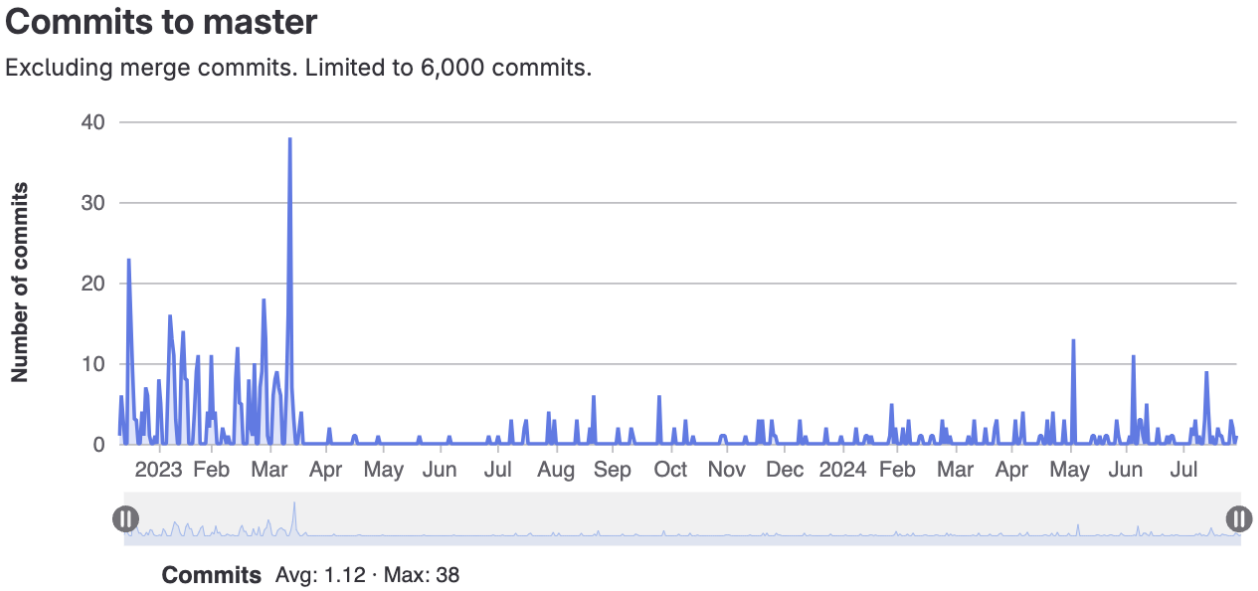
Во втором проекте DBT созданий view не отрабатывает за час
В коде витрин много CTE

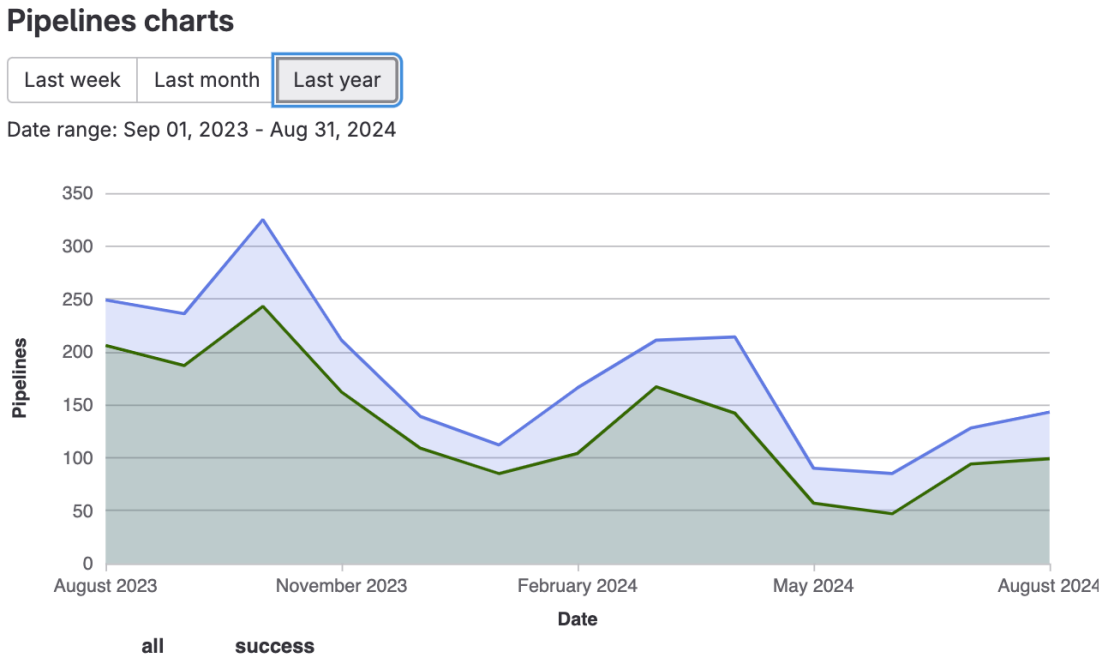
схема копия прода но с небольшим куском данных
Создаем не view а настоящую таблицу

Но в первом проекте оказалось что быстрее работает создание view

08 Подводим итоги

Витрины спустя 2 года





Высокий темп изменений

- project_1
  - Finished running 85 view models in 0 hours 11 minutes and 31.05 seconds (691.05s).
- project_2
  - Finished running 33 table models, 8 view models in 0 hours 14 minutes and 16.70 seconds (856.70s).
- project_3
  - Found 106 models, 6 sources

О команде 2022

6 человек

OMNI

Snowplow (Уход от Google Analytics)

Team Lead / 1

DA / 1

DA / 1

Head DE / 1

DE / 1

DE / 1

О команде 2024

25 человек

	OMNI	Snowplow (Уход от Google Analytics)	CVM	Platform
Team Lead / 2	DA / 2	DA / 2	DA / 3	DA / 1
Head DS / 1			DS / 6	
Head DE / 1	DE / 2	DE / 1	DE / 3	DE / 1

Спасибо!