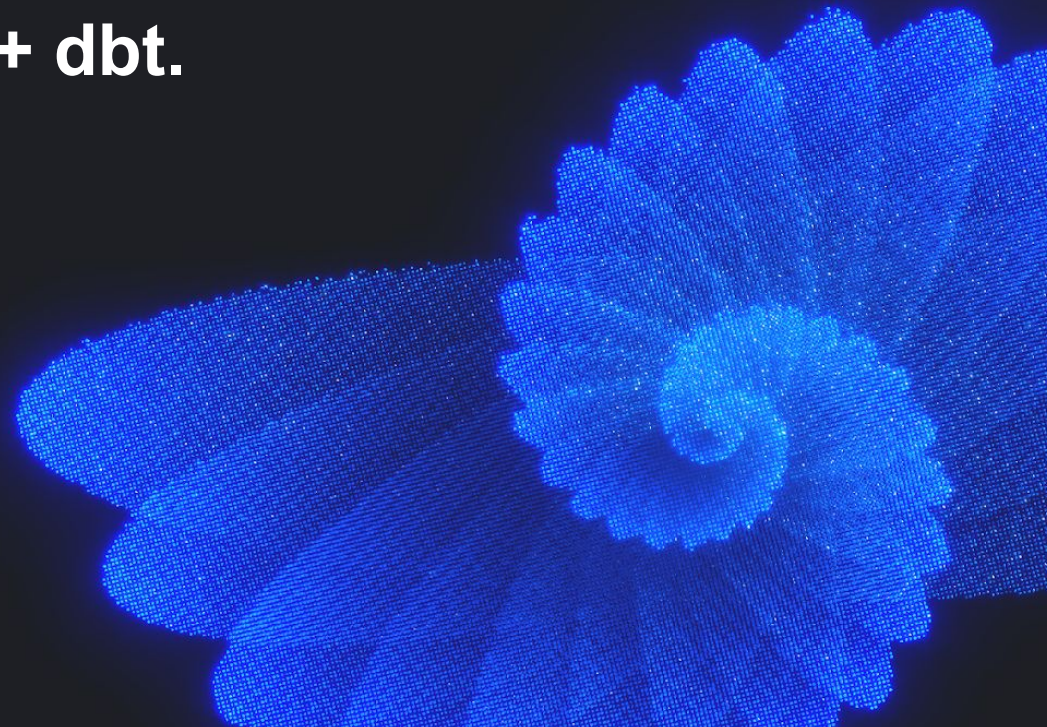
Toloka

От хайпа до продакшена: DataMesh на Airflow + dbt.



@nikitayurasov
@lkozhinov

Никита Юрасов

Разработчик по призванию, data engineer по профессии и математик в душе.

Кожинов Леонид

Старший разработчик мигрировавший в DevOps’а. Фанат модных языков программирования, если это C++.

Какой план?

- О нас
- Инфраструктура
- Data Mesh

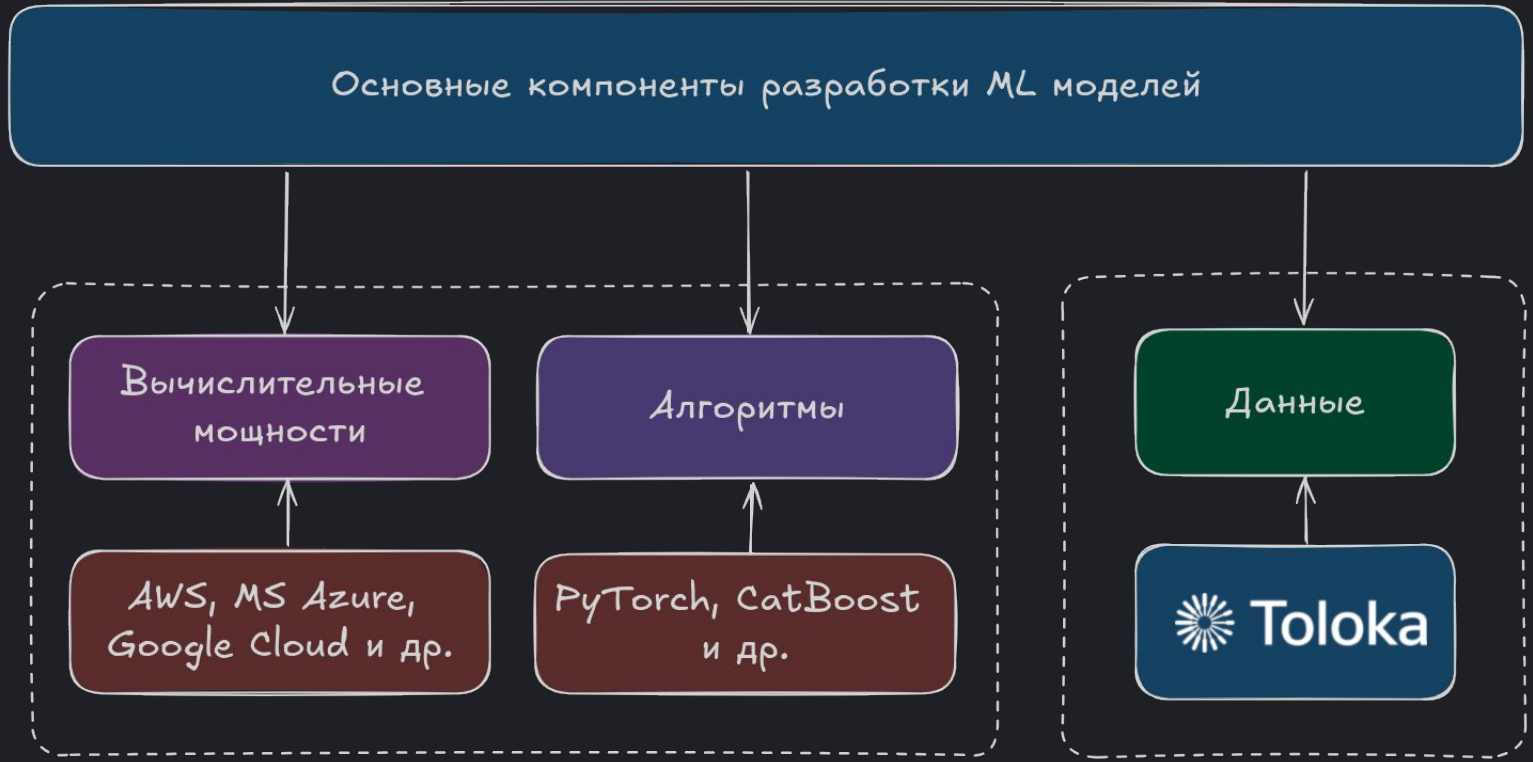
- dbt + Airflow
- dbt-af = ❤️

- Управление нагрузкой
- Data quality
- Development experience

- Плюсы/минусы
- Боли
- Выводы

Глава I. Будем знакомы

Данные - это новая нефть ©



https://toloka.ai/about-us/

Наша прекрасная Data и ML платформа!

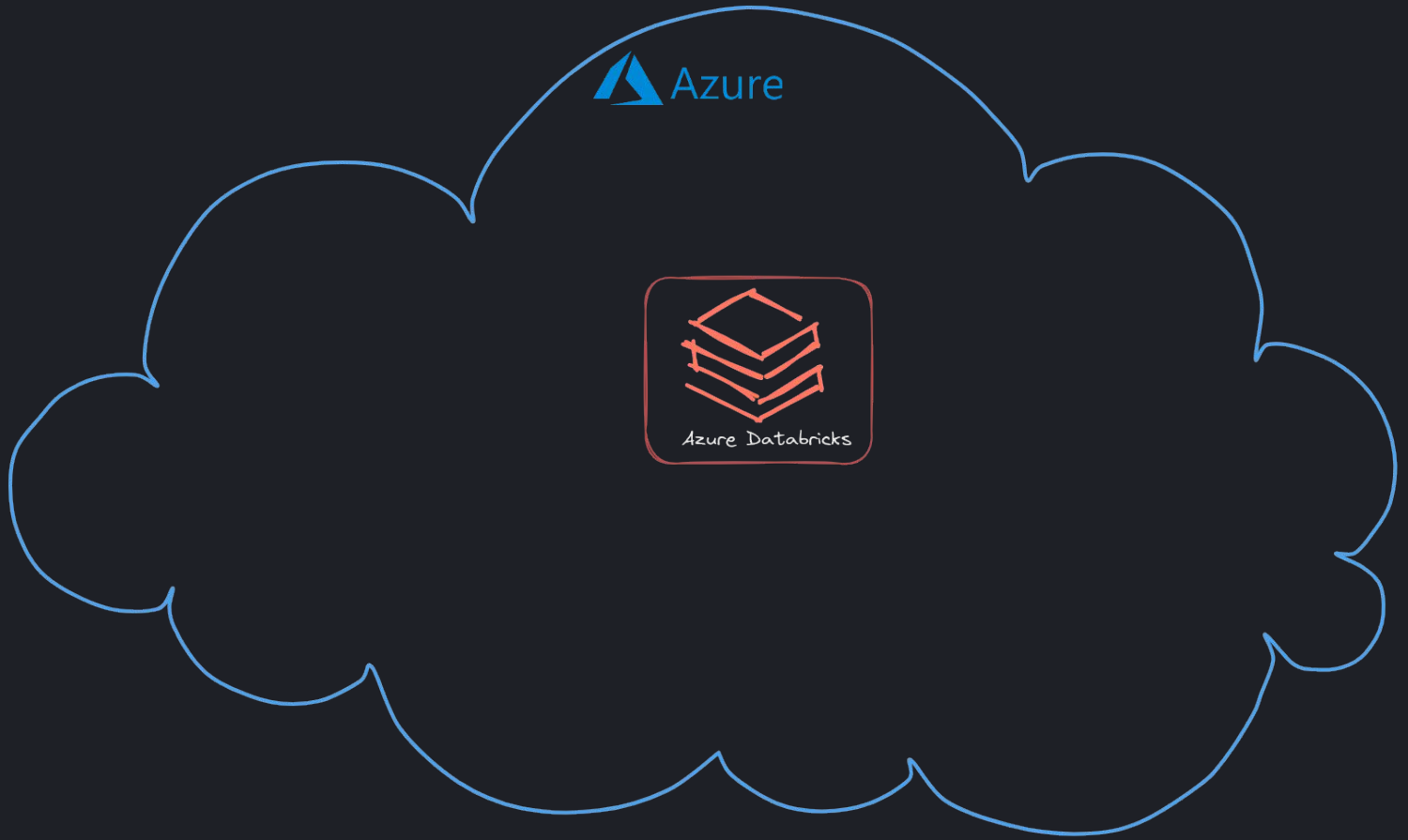
530 Тb

400 CPU

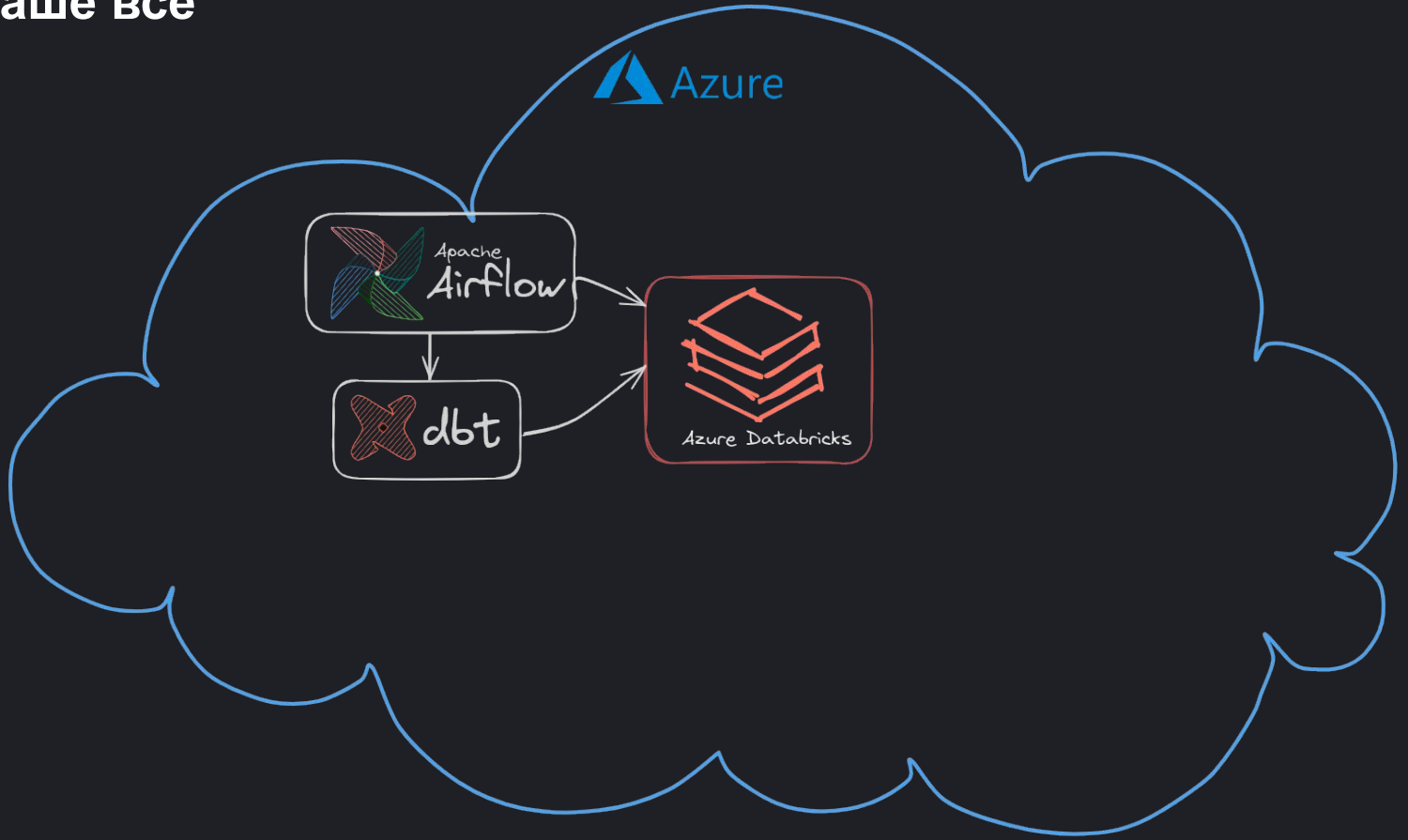
4700 таблиц

100 внутренних пользователей

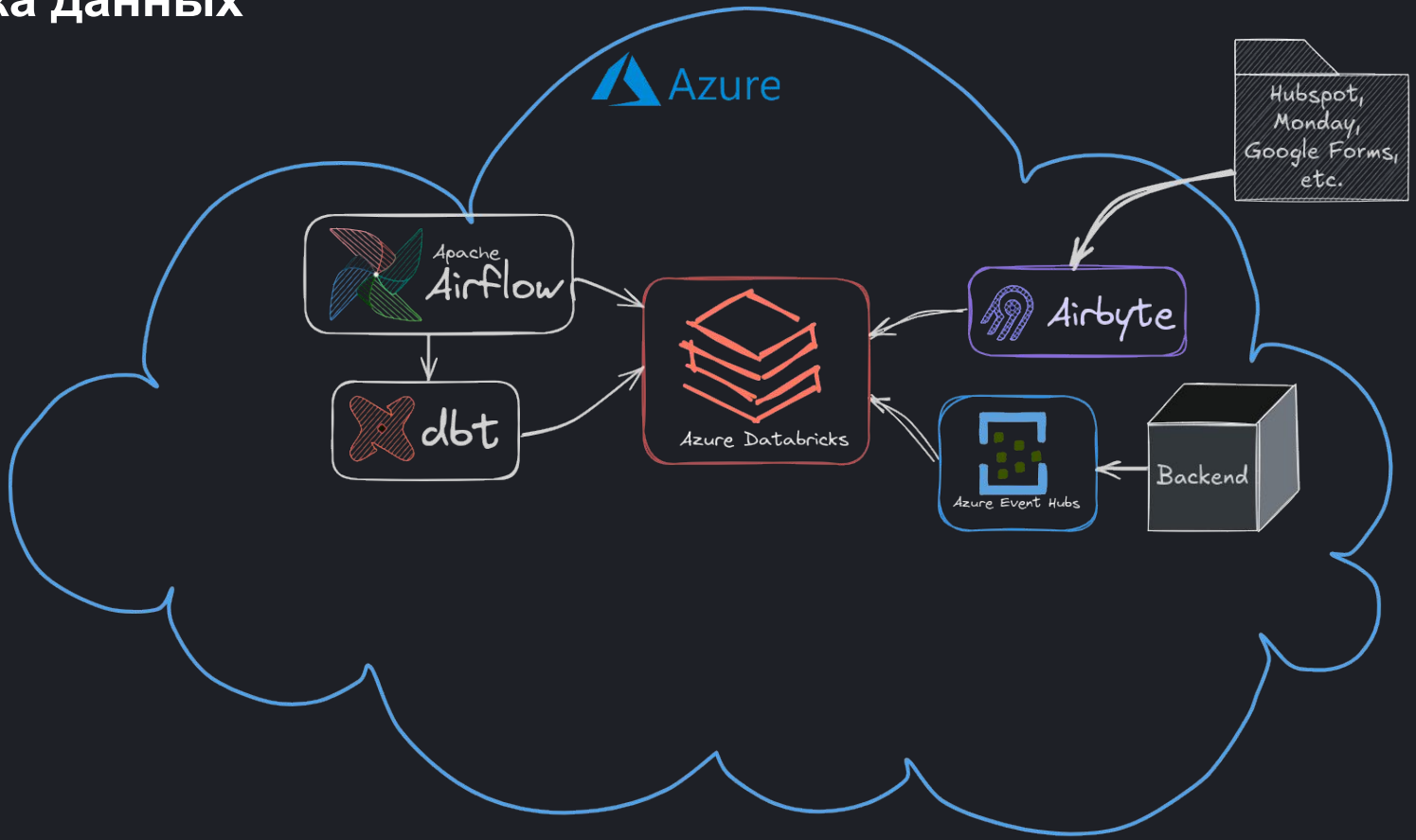
Databricks - ядро нашего DataLake house’a



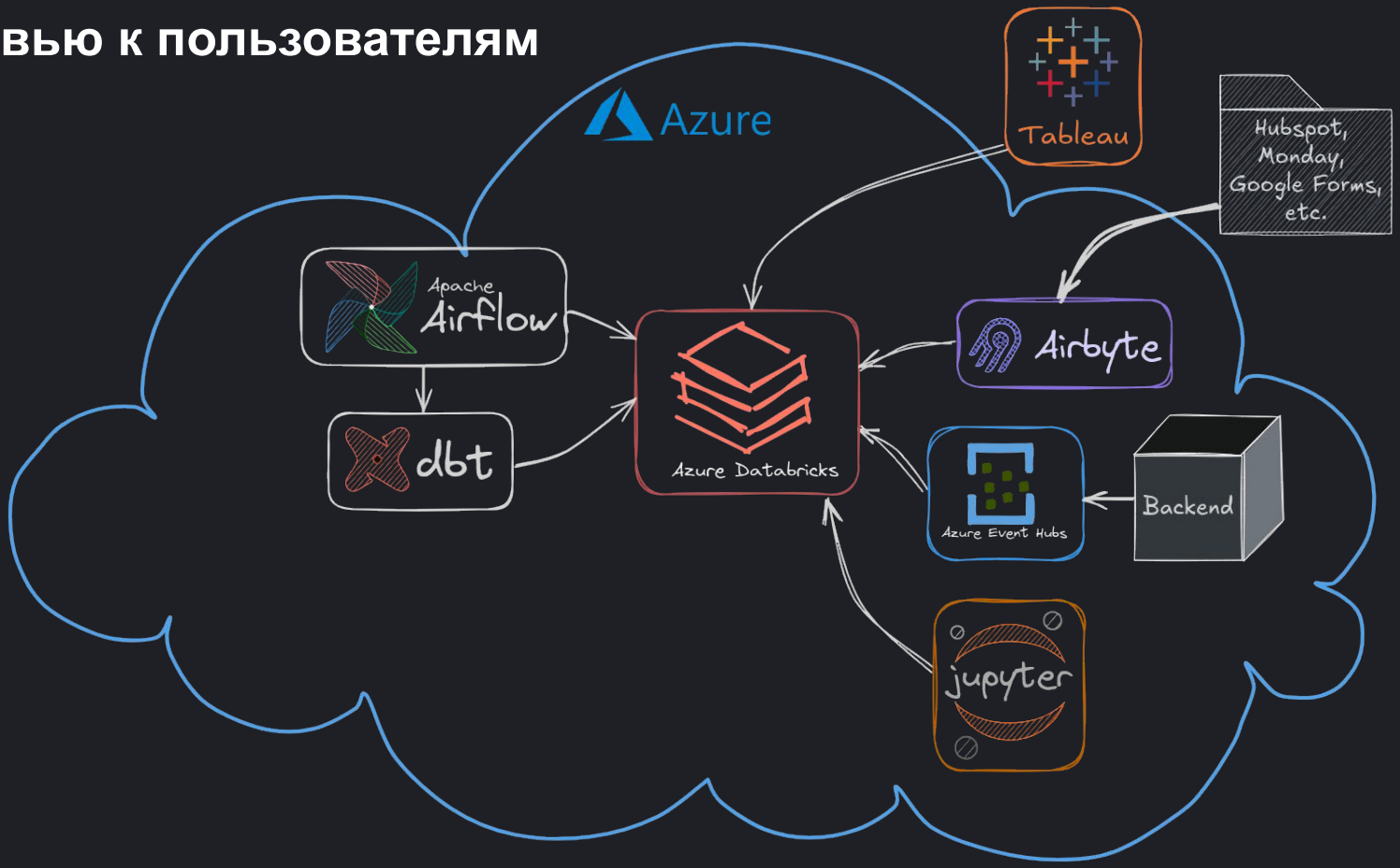
DBT - наше всё



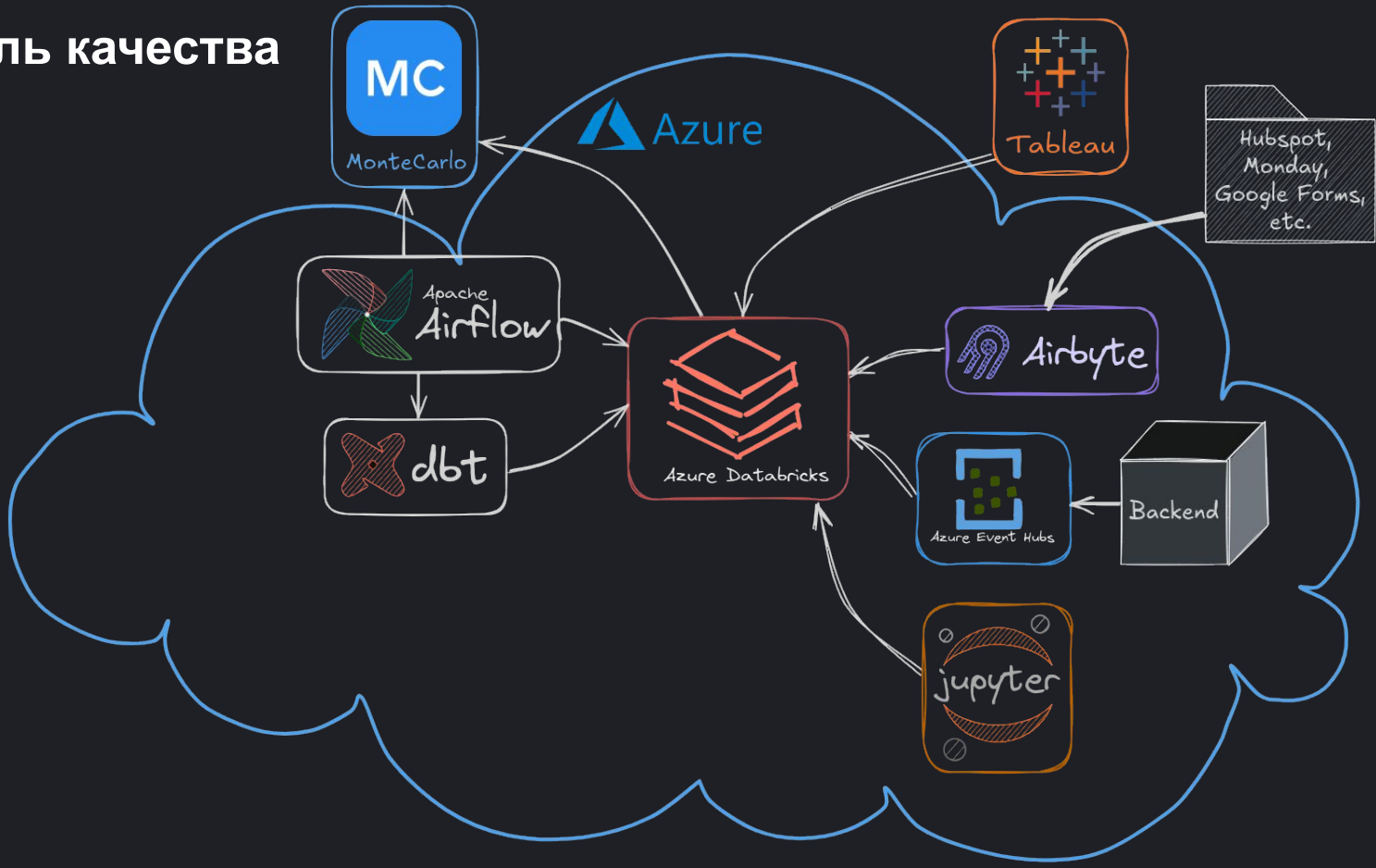
Загрузка данных



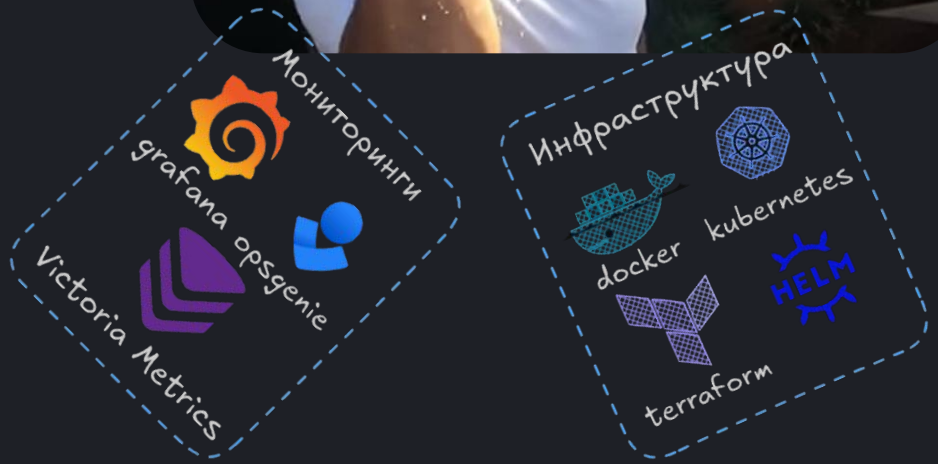
С любовью к пользователям



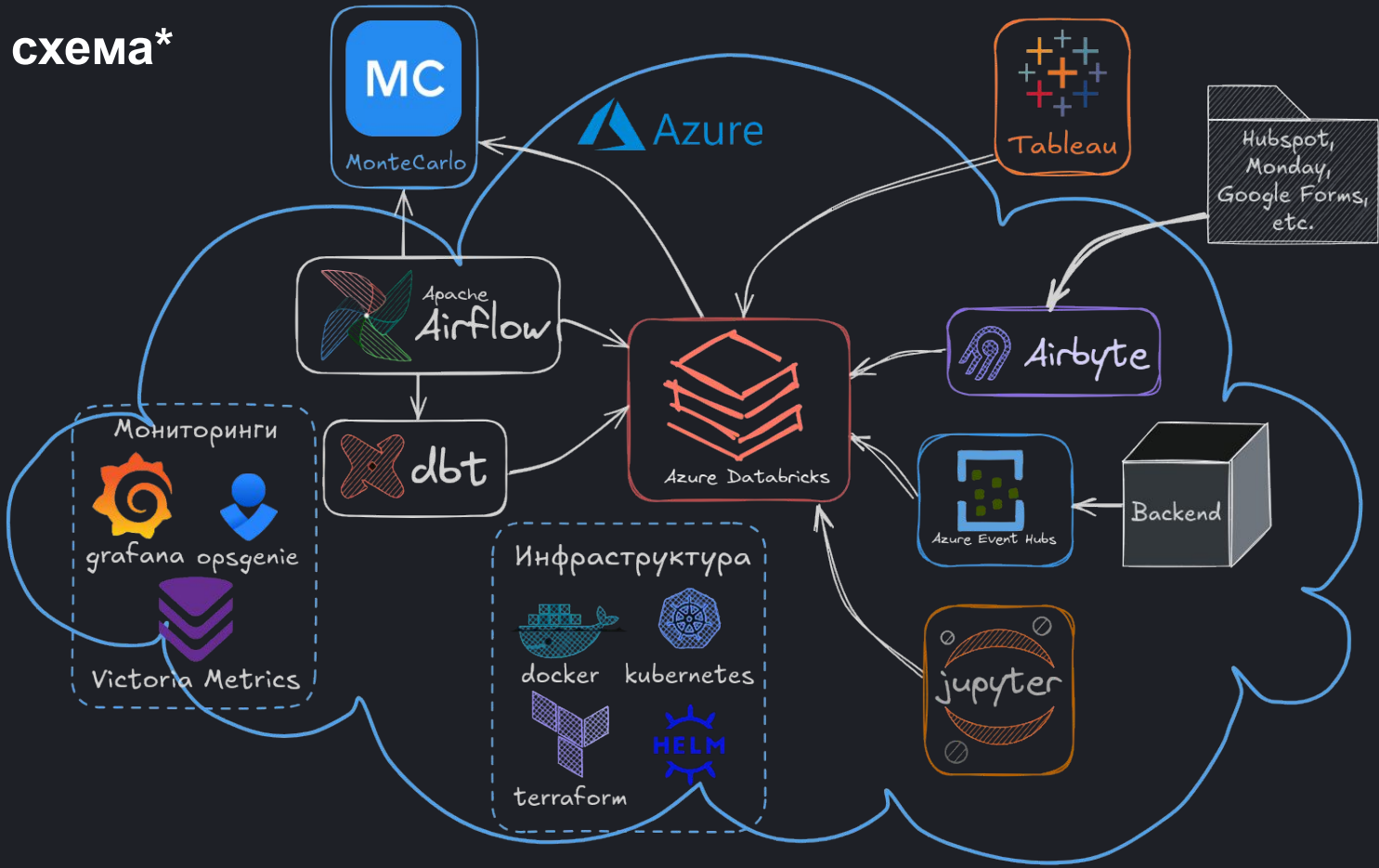
Контроль качества



Приправим мониторингами и инфрой

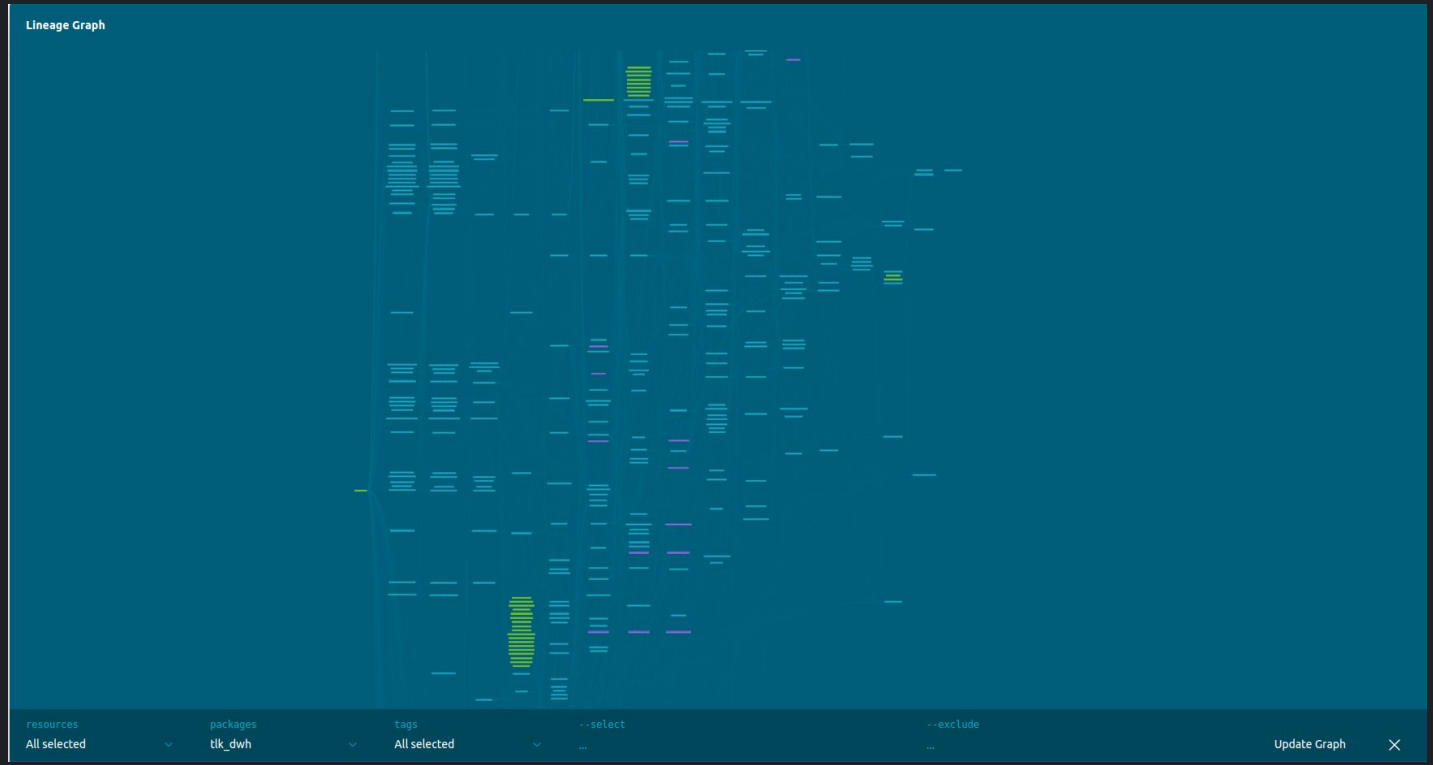


Полная схема*



Так что, dbt run и “поехали”, да?

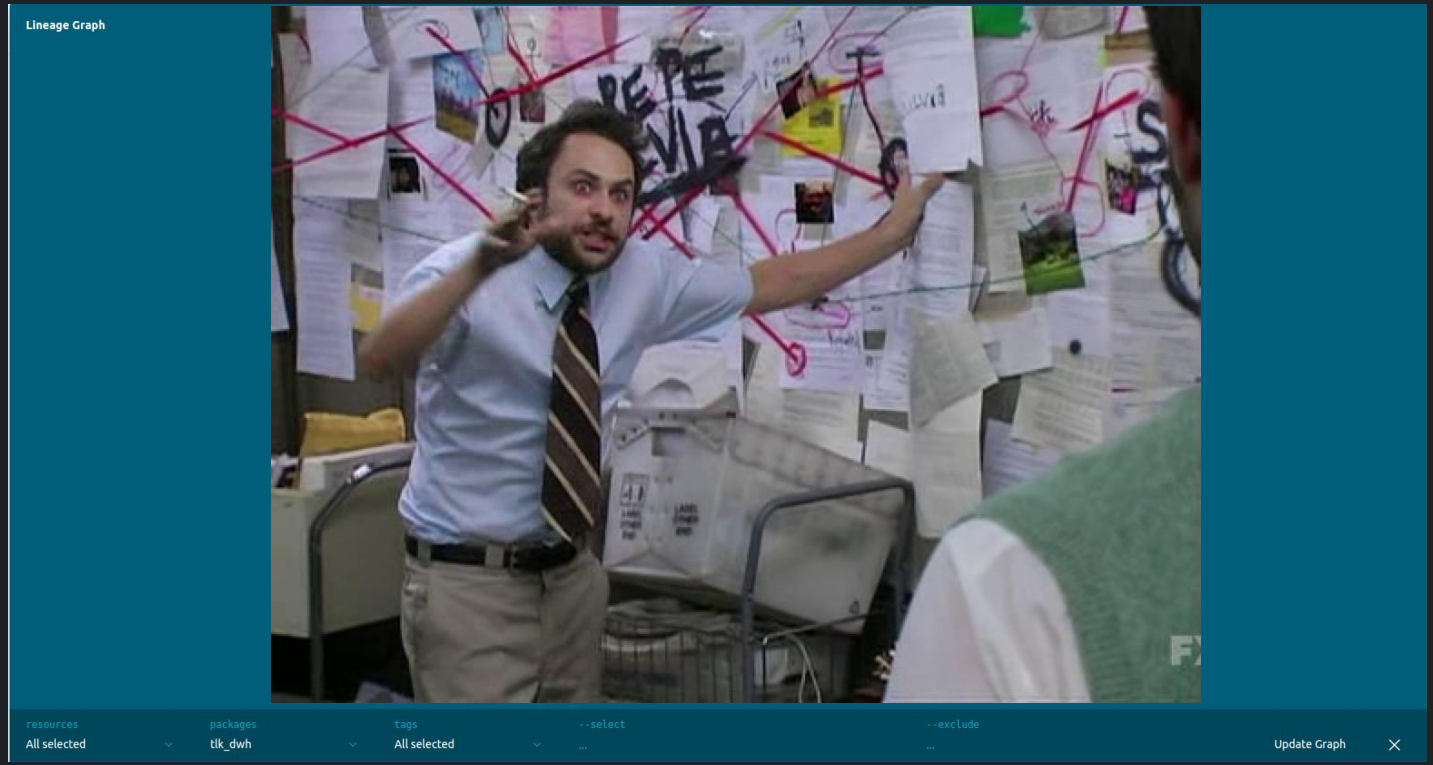
Размер имеет значение



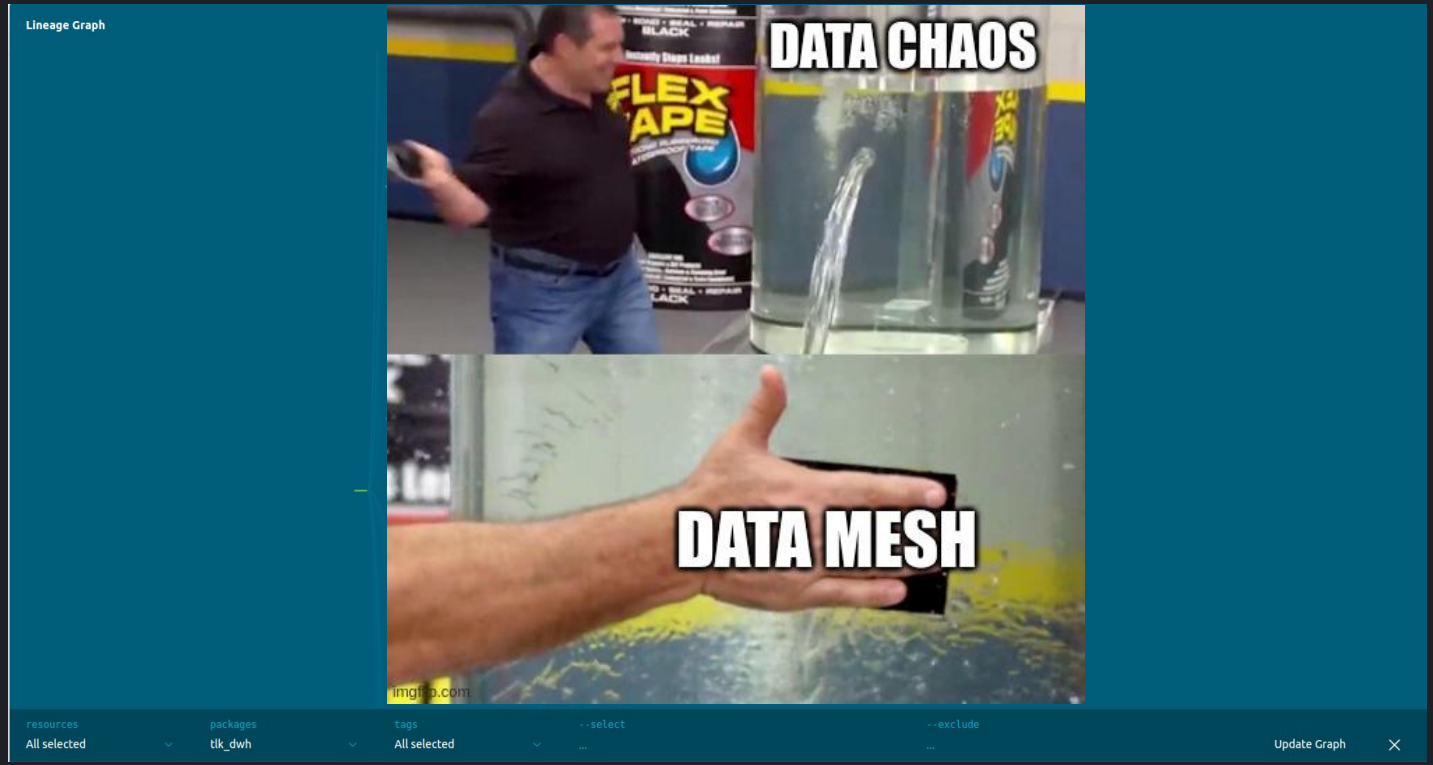
Toasty!


Lineage Graph
EventHub
resources
All selected
packages
tlk_dwh
tags
All selected
--select
...
--exclude
...

Бесконечность – не предел ©



И сказал лид “Декомпозируй” и стало хорошо



Домен 1

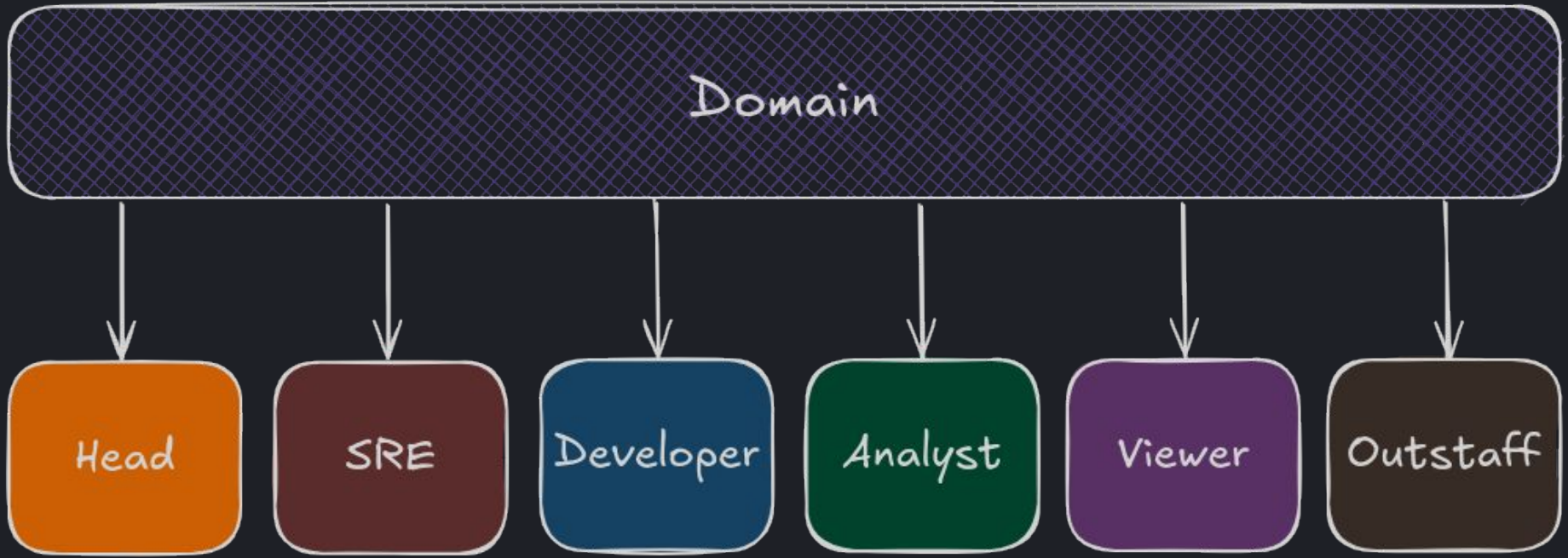
- Название: svc_crm
- Ответственные: …
- Доменные RBAC роли

Toloka

Домен 2

- Название: dmn_workers
- Ответственные: …
- Доменные RBAC роли

Доменные роли

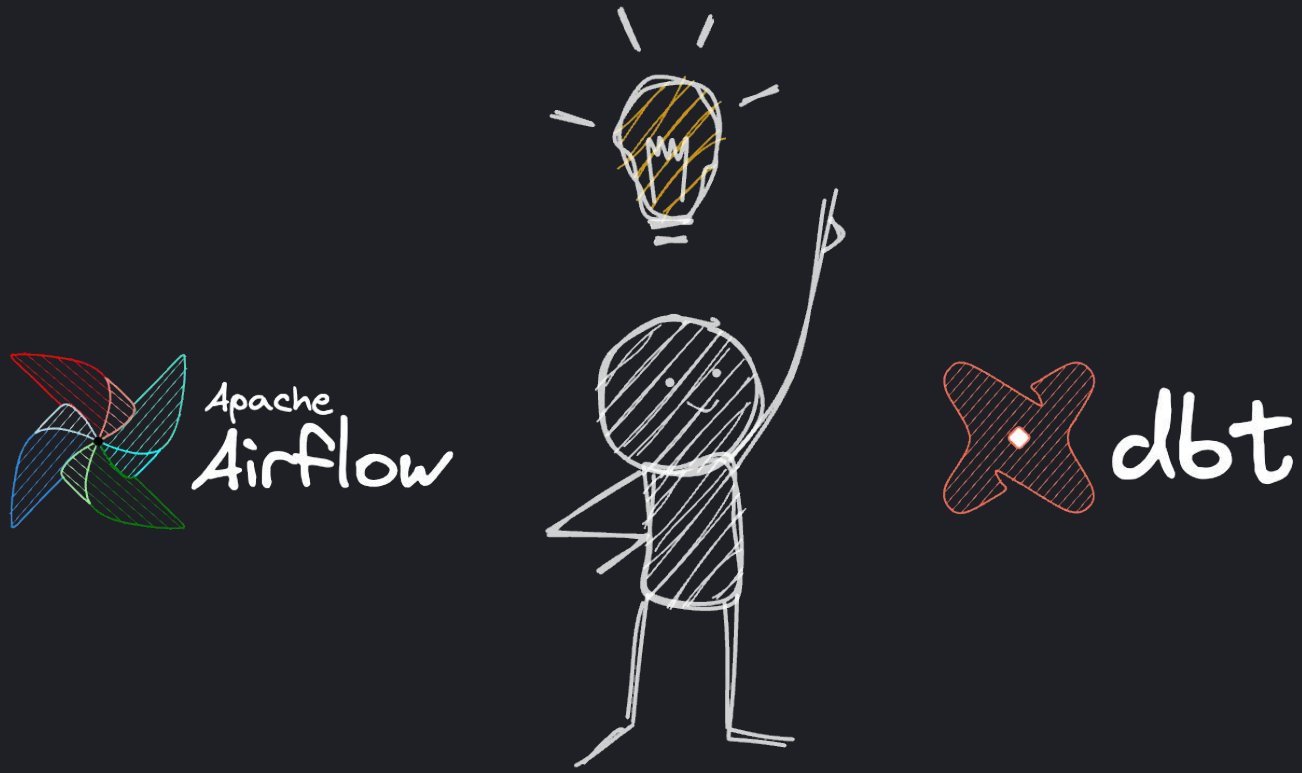


Интеграция ролевой модели с сервисами

	Azure Databricks (data)	Azure Databricks (compute)	Apache Airflow	GitHub	Slack	Tableau	Airbyte
SRE	Полный доступ	Полный доступ	Полный доступ	Полный доступ	Группа sre	-	
Developer	Чтение	Общий кластер	Полный доступ	Полный доступ	Группа developer	-	
Analyst	Чтение	Доменный кластер	Чтение	Чтение	-	Полный доступ	
Viewer	Чтение	Общий кластер	-	-	-	-	
Outstaff	Чтение	Общий кластер	-	-	-	-	-
Head	-	-	-	-	Группа head	-	

Последний босс

Глава II. Решающая



Что мы хотим?

- Авто-генерация DAG
- Идемпотентность запусков dbt
- Поддержка разных уровней шедулинга

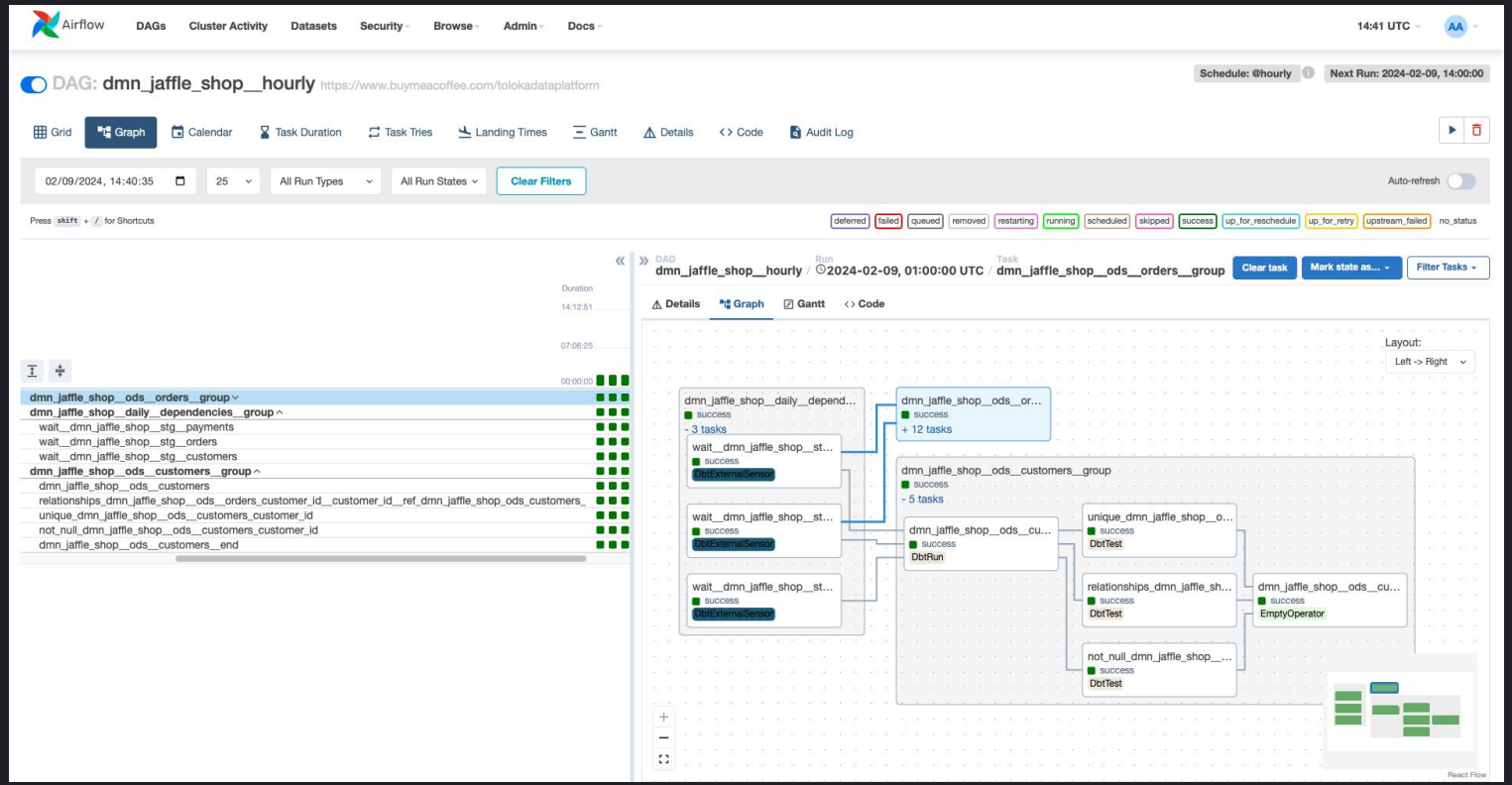
Дополнительно:

- Доменно-ориентированность (Data Mesh)
- Интеграции с другими системами (Airbyte, Monte Carlo, …)

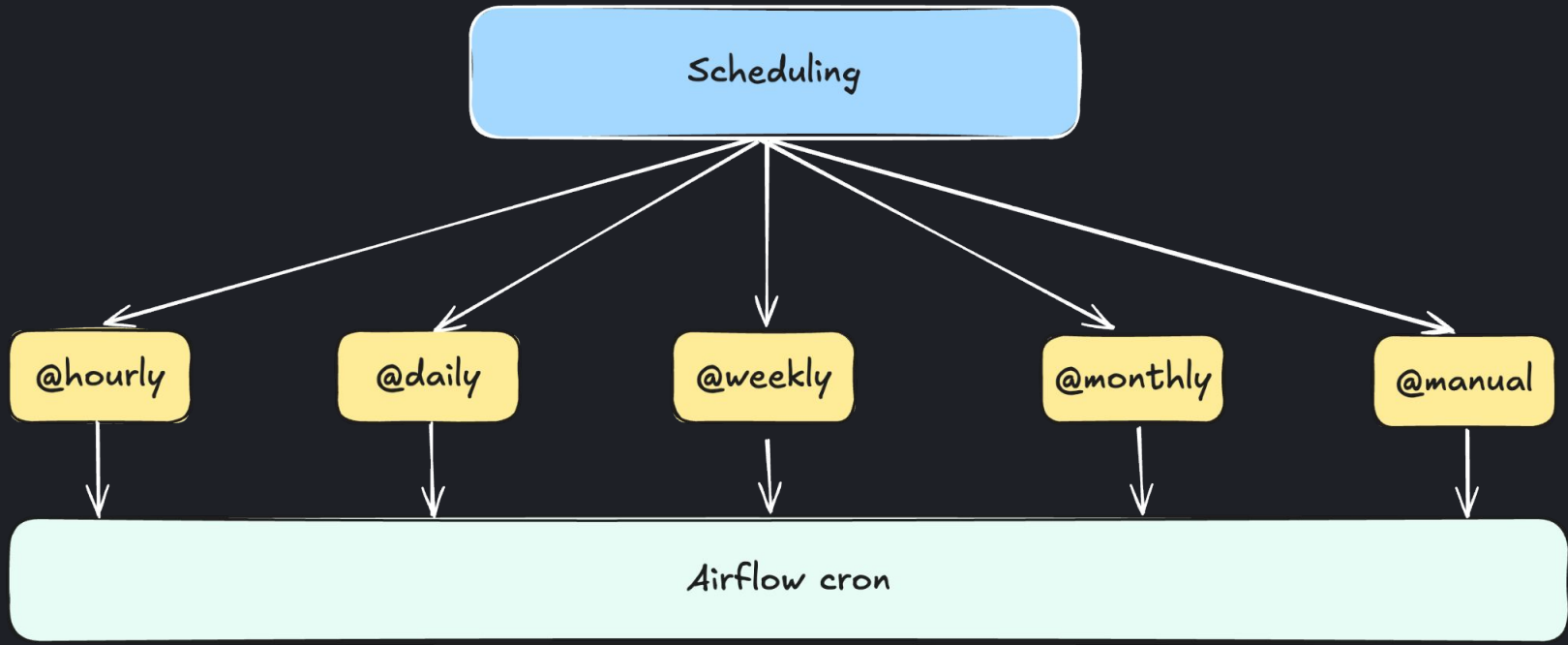
Велосипеды

	astronomer-cosmos	dbt-airflow	dbt Cloud	Dagster
Авто-генерация DAG'ов	+/-	+/-	-	++
Идемпотентность запусков	+	+	+	+
Разные уровни шедулингв	-	-	-	-/+
Управление нагрузкой, сценарии тестов	-	-	-	-/+

dbt-af


Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
14:41 UTC
AA
DAG: dmn_jaffle_shop__hourly https://www.buymeacoffee.com/tolokadataplatform
Schedule: @hourly
Next Run: 2024-02-09, 14:00:00
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
02/09/2024, 14:40:35
25
All Run Types
All Run States
Clear Filters
Auto-refresh
Press shift + / for Shortcuts
deferred
failed
queued
removed
restarting
running
scheduled
skipped
success
up_for_reschedule
up_for_retry
upstream_failed
no_status
Duration
14:12:51
07:06:25
00:00:00
dmn_jaffle_shop__ods__orders__group
dmn_jaffle_shop__daily__dependencies__group
wait__dmn_jaffle_shop__stg__payments
wait__dmn_jaffle_shop__stg__orders
wait__dmn_jaffle_shop__stg__customers
dmn_jaffle_shop__ods__customers__group
dmn_jaffle_shop__ods__customers
relationships_dmn_jaffle_shop__ods__orders_customer_id__customer_id__ref_dmn_jaffle_shop_ods_customers_
unique_dmn_jaffle_shop__ods__customers_customer_id
not_null_dmn_jaffle_shop__ods__customers_customer_id
dmn_jaffle_shop__ods__customers__end
DAG
dmn_jaffle_shop__hourly
Run
2024-02-09, 01:00:00 UTC
Task
dmn_jaffle_shop__ods__orders__group
Clear task
Mark state as...
Filter Tasks
Details
Graph
Gantt
Code
Layout:
Left -> Right
dmn_jaffle_shop__daily__depend...
success
- 3 tasks
wait__dmn_jaffle_shop__st...
success
DbtExternalSensor
wait__dmn_jaffle_shop__st...
success
DbtExternalSensor
wait__dmn_jaffle_shop__st...
success
DbtExternalSensor
dmn_jaffle_shop__ods__or...
success
+ 12 tasks
dmn_jaffle_shop__ods__customers__group
success
- 5 tasks
dmn_jaffle_shop__ods__cu...
success
DbtRun
unique_dmn_jaffle_shop__o...
success
DbtTest
relationships_dmn_jaffle_sh...
success
DbtTest
not_null_dmn_jaffle_shop__...
success
DbtTest
dmn_jaffle_shop__ods__cu...
success
EmptyOperator
React Flow

Шедулинг (от @monthly до @hourly)

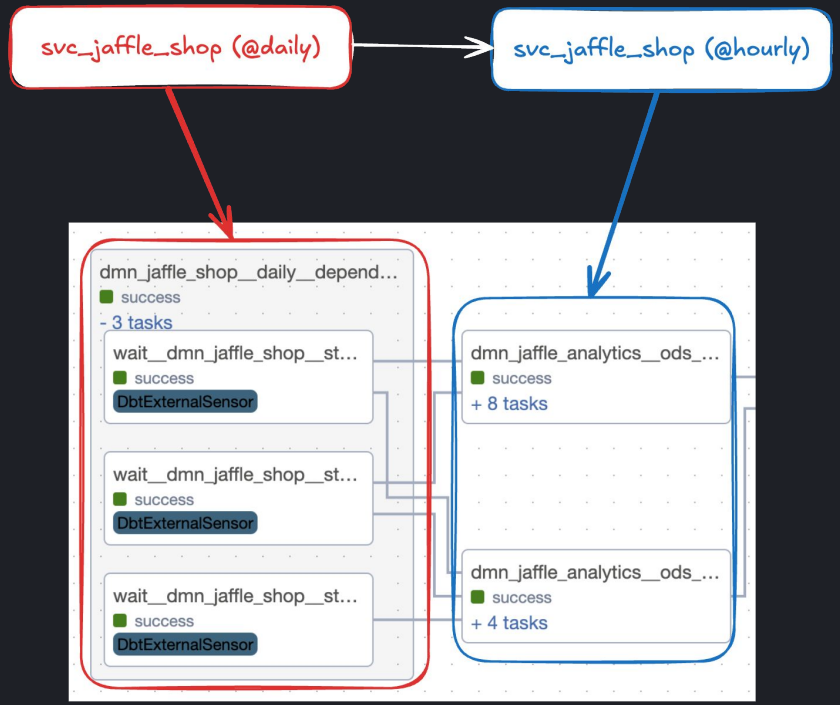


Кросс-доменные зависимости

- Зависимости указаны через функцию **ref**
- Не создается циклический граф из DAG’ов

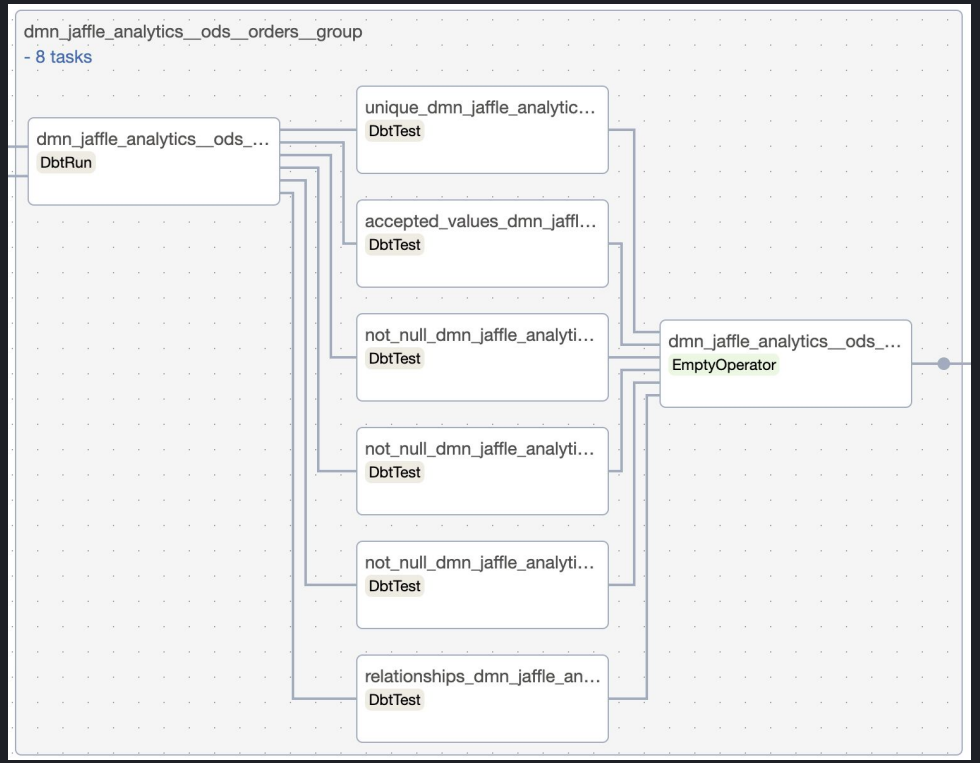
Кросс-доменные зависимости

- Зависимости указаны через функцию **ref**
- Не создается циклический граф из DAG’ов



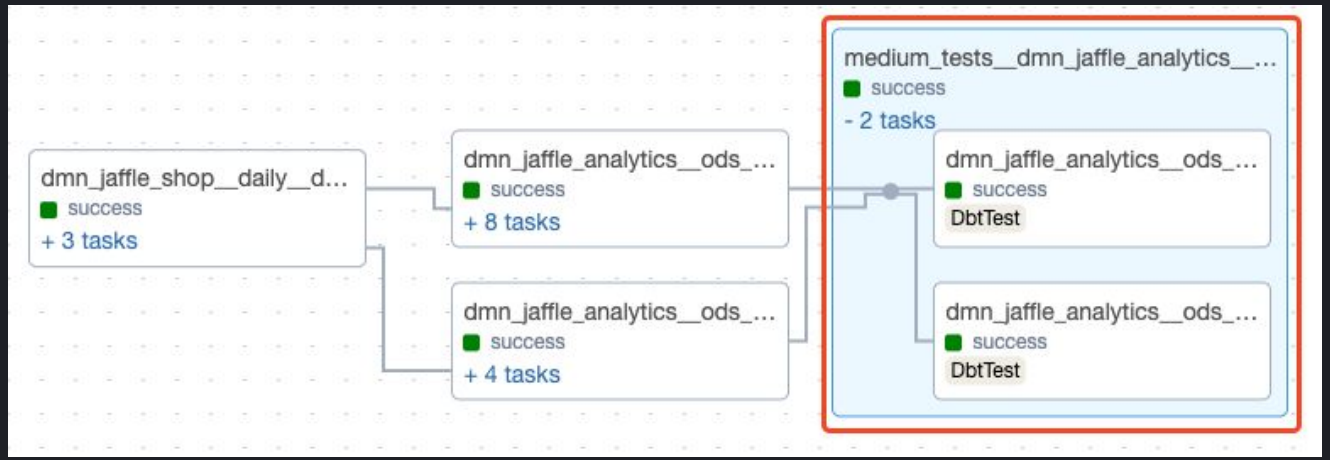
3 сценария запуска тестов

- @small
  - **легкие**
  - **блокируют downstream**
- @medium
- @large



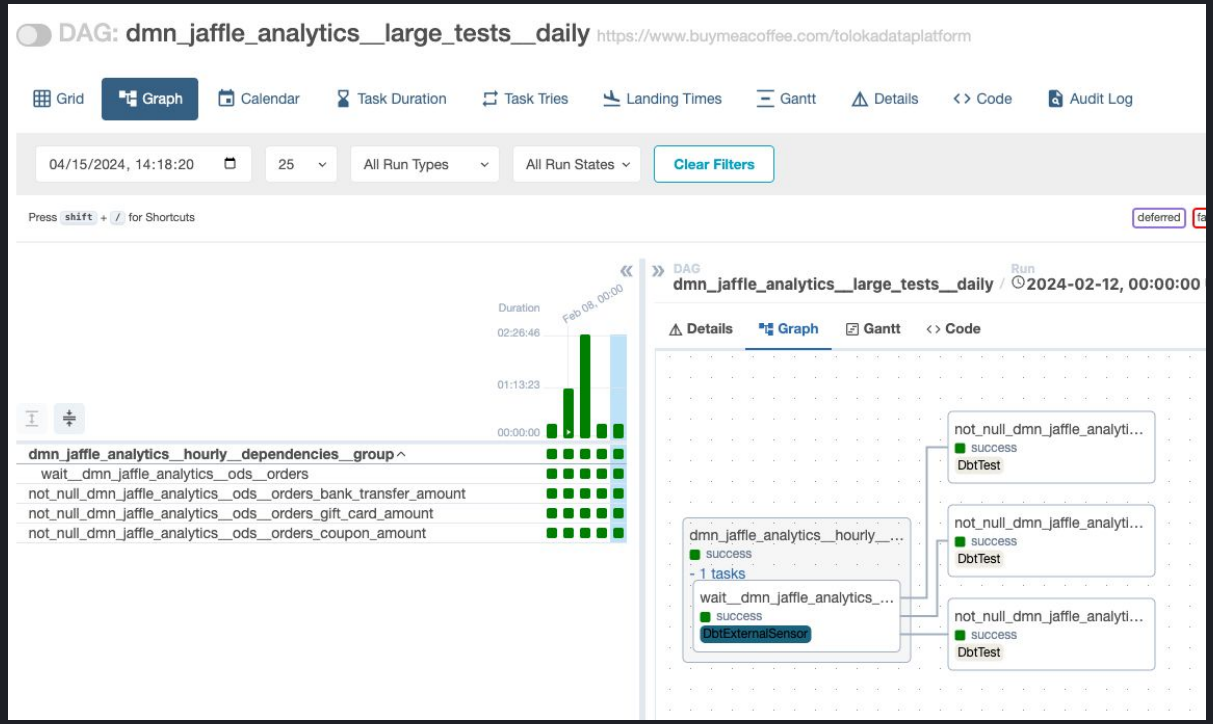
3 сценария запуска тестов

- @small
- @medium
  - **Запускаются в конце DAG Run**
- @large



3 сценария запуска тестов

- @small
- @medium
- @large
  - **особо тяжелые тесты**
  - **отдельный DAG**

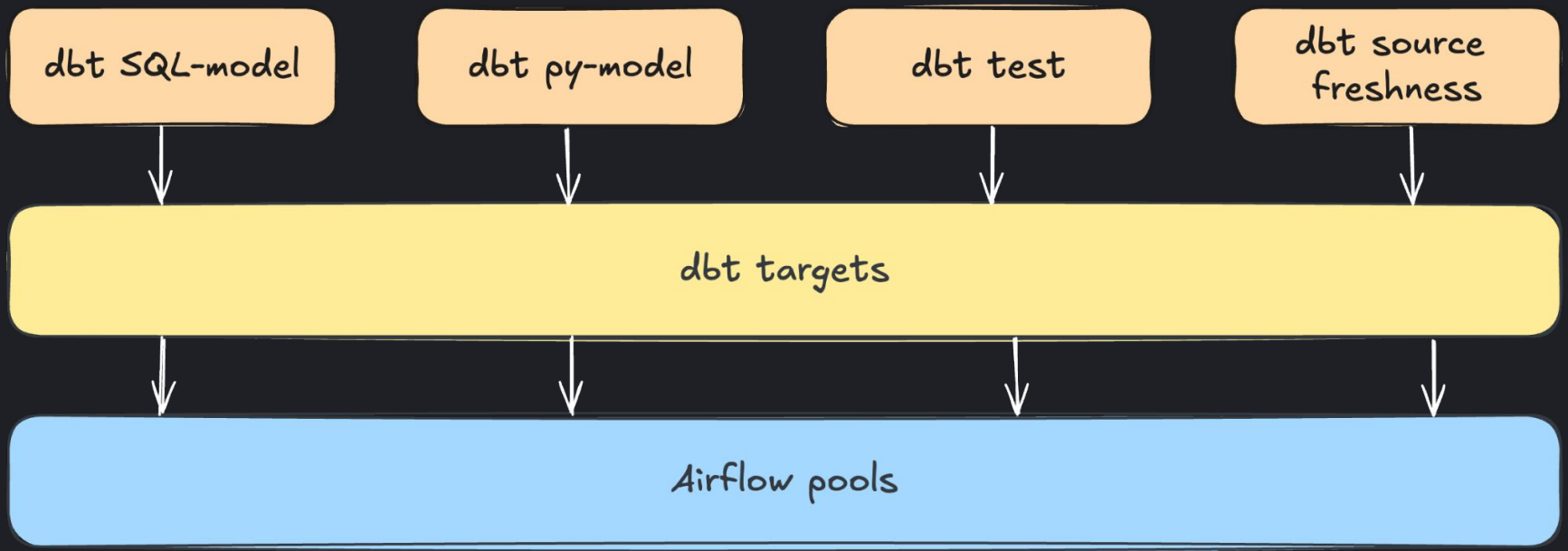


Дополнительные фичи

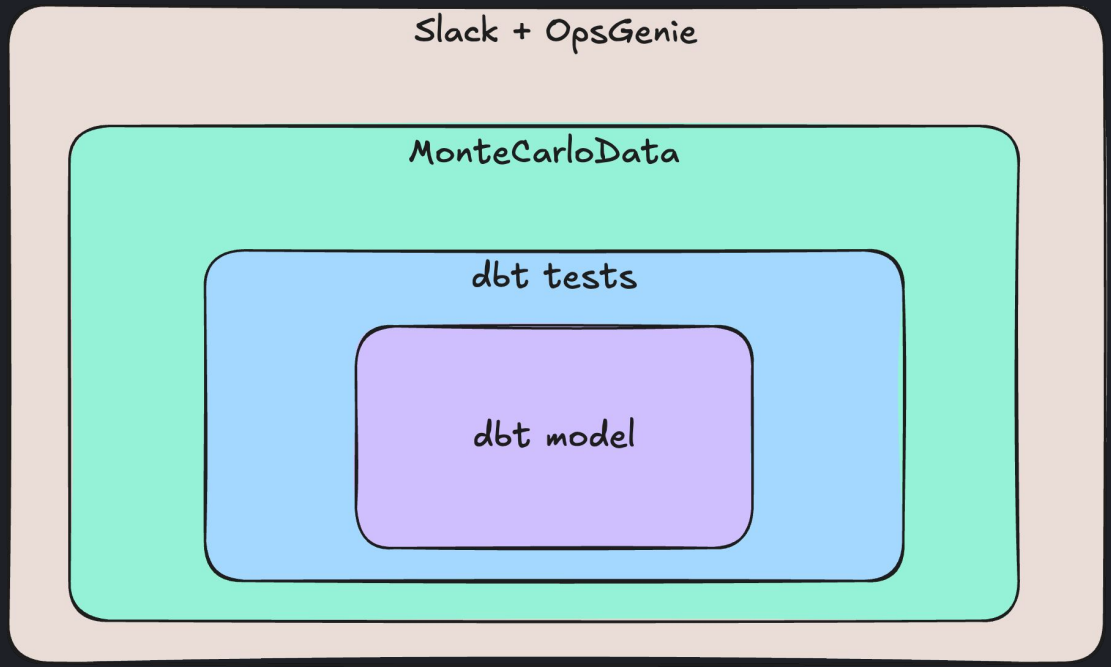
- explicit dbt targets
- source freshness
- Интеграция с другими системами (Monte Carlo, Tableau)
- произвольные k8s dbt-python-like таски
- minidbt (оптимизация компиляции dbt проекта)

Глава III. dbt-af в дикой природе

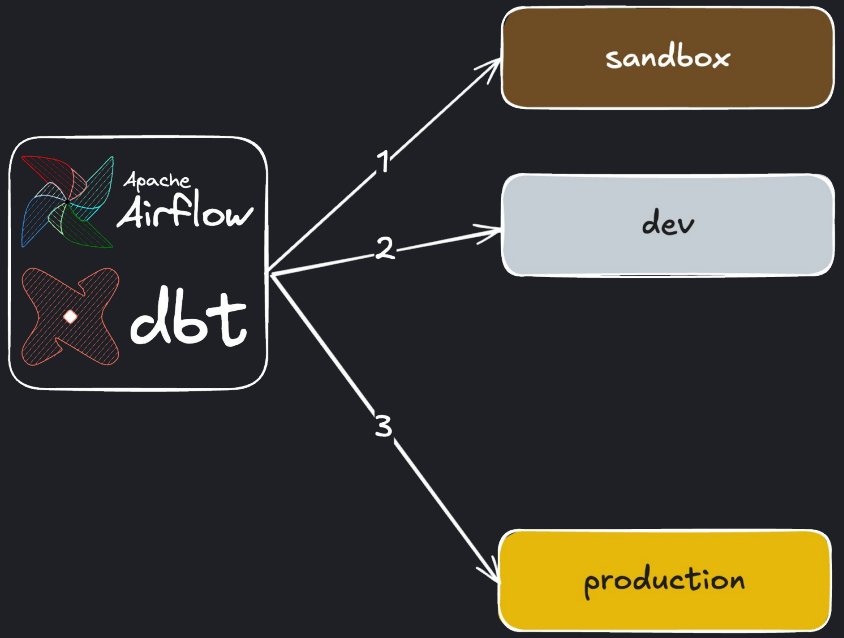
С большим продом приходит большая ответственность



DQ: как мы узнаем, что оно развалилось



DX дата инженера



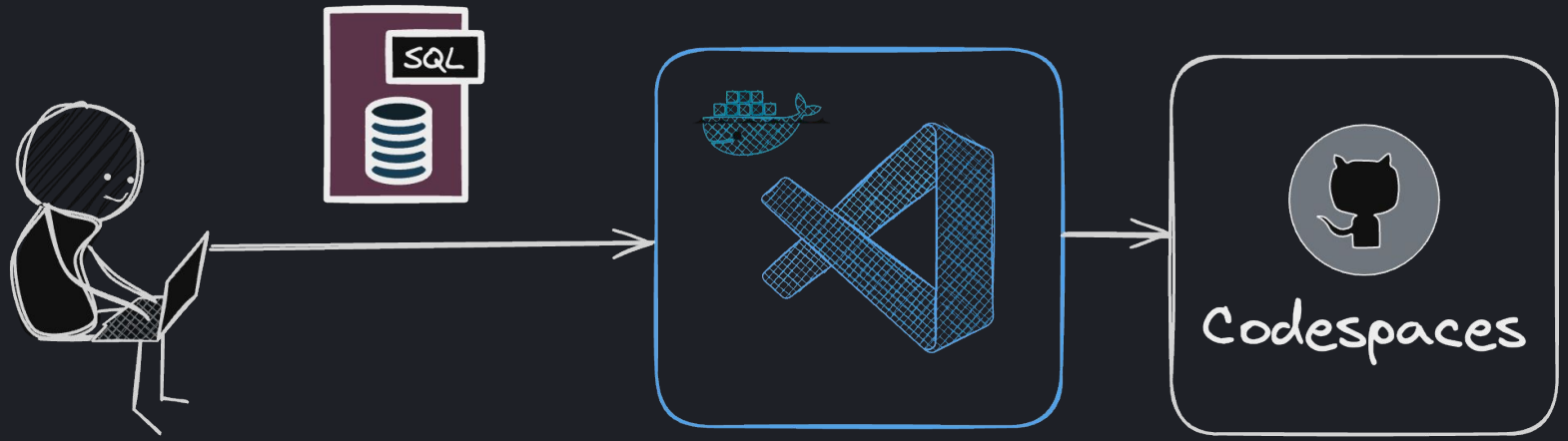

```python
# LABELS: dag, airflow (it's required for airflow dag-processor)
from dbt_af.dags import compile_dbt_af_dags
from dbt_af.conf import Config, DbtDefaultTargetsConfig,
DbtProjectConfig
# specify here all settings for your dbt project
config = Config(
    dbt_project=DbtProjectConfig(
        dbt_project_name='my_dbt_project',
        dbt_project_path='/path/to/my_dbt_project',
        dbt_models_path='/path/to/my_dbt_project/models',
        dbt_profiles_path='/path/to/my_dbt_project',
        dbt_target_path='/path/to/my_dbt_project/target',
        dbt_log_path='/path/to/my_dbt_project/logs',
        dbt_schema='my_dbt_schema',
    ),
    dbt_default_targets=DbtDefaultTargetsConfig(default_target='dev'),
    is_dev=False,  # set to True if you want to turn on dry-run mode
)

dags = compile_dbt_af_dags(
    manifest_path='/path/to/my_dbt_project/target/manifest.json',
    config=config,
)
for dag_name, dag in dags.items():
    globals()[dag_name] = dag
```

DX аналитика

```
models:
    - name: "domain.schema.table"
      description: "Some meaningful description"
      config:
        schedule: "@hourly"
        dbt_target: "nightly_target"
        dependencies:
            other_domain.other_schema.other_table:
                skip: true
```

Просто добавь контейнер

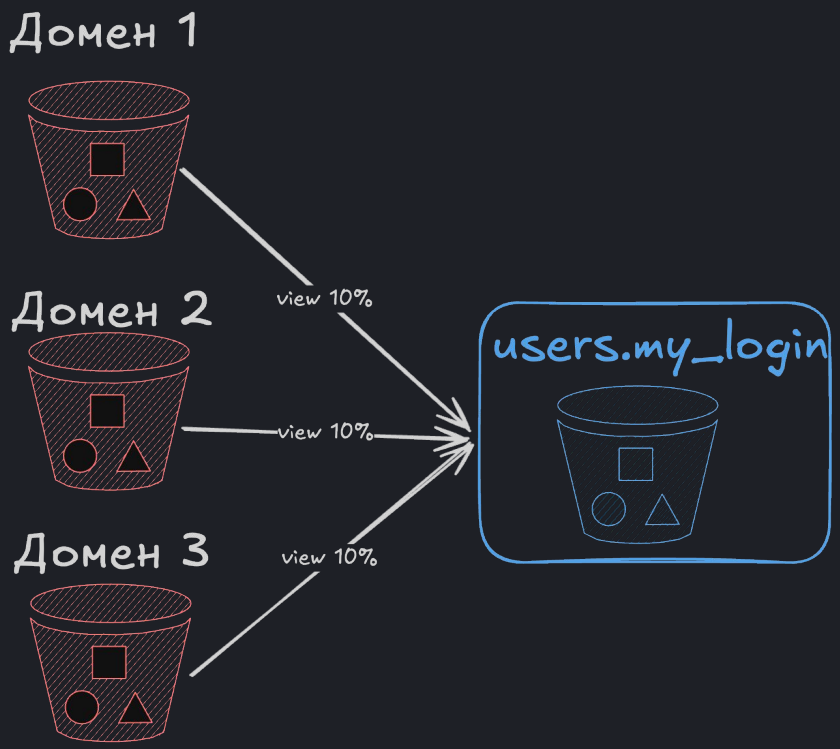


Мы не тестируем на проде

```
-- dbt/models/domain/layer/some_table.sql
{{
    config(
        materialized="table",
        schedule="@hourly"
    )
}}
select *
from {{ dbt.ref('other_domain.layer.table_name') }}
```

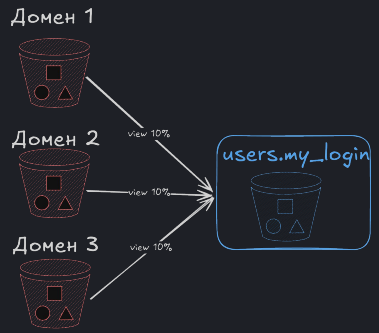
Мы не тестируем на проде

1. Сэмплирование



Мы не тестируем на проде

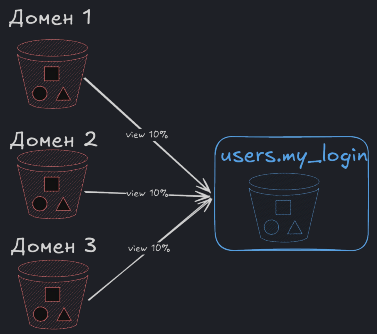
1. Сэмплирование



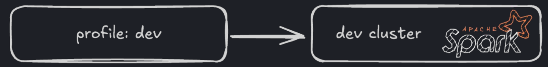
2. dbt run --target

Мы не тестируем на проде

1. Сэмплирование



2. dbt run --target



3. Магия макросов

```
{{ dbt.ref('other_domain.layer.table_name') }}
```

```
`users`.`my_login`.`other_domain__layer__table_name`
```

Мы не тестируем на проде

4. Компиляция

```
-- dbt/models/domain/layer/some_table.sql
{{
    config(
        materialized="table",
        schedule="@hourly"
    )
}}
select *
from {{ dbt.ref('other_domain.layer.table_name') }}
```

```
create or replace table `users`.`my_login`.`domain__layer__some_table`
as
select *
from `users`.`my_login`.`other_domain__layer__table_name`
```

Статические тесты

```
-- используй dbt.ref
✅ from {{ dbt.ref('domain.layer.table') }}
❌ from `domain`.`layer`.`table`

-- контракт на нейминг
✅ domain.layer.table
❌ domain__layer.table
```

Глава IV. Заключительная

Немного статистики

- 32 доменов
- ~8900 запусков dbt run в сутки
- ~310 уникальных связей между доменами
- ~320 тестов
- 59k строк sql-кода

Наши боли

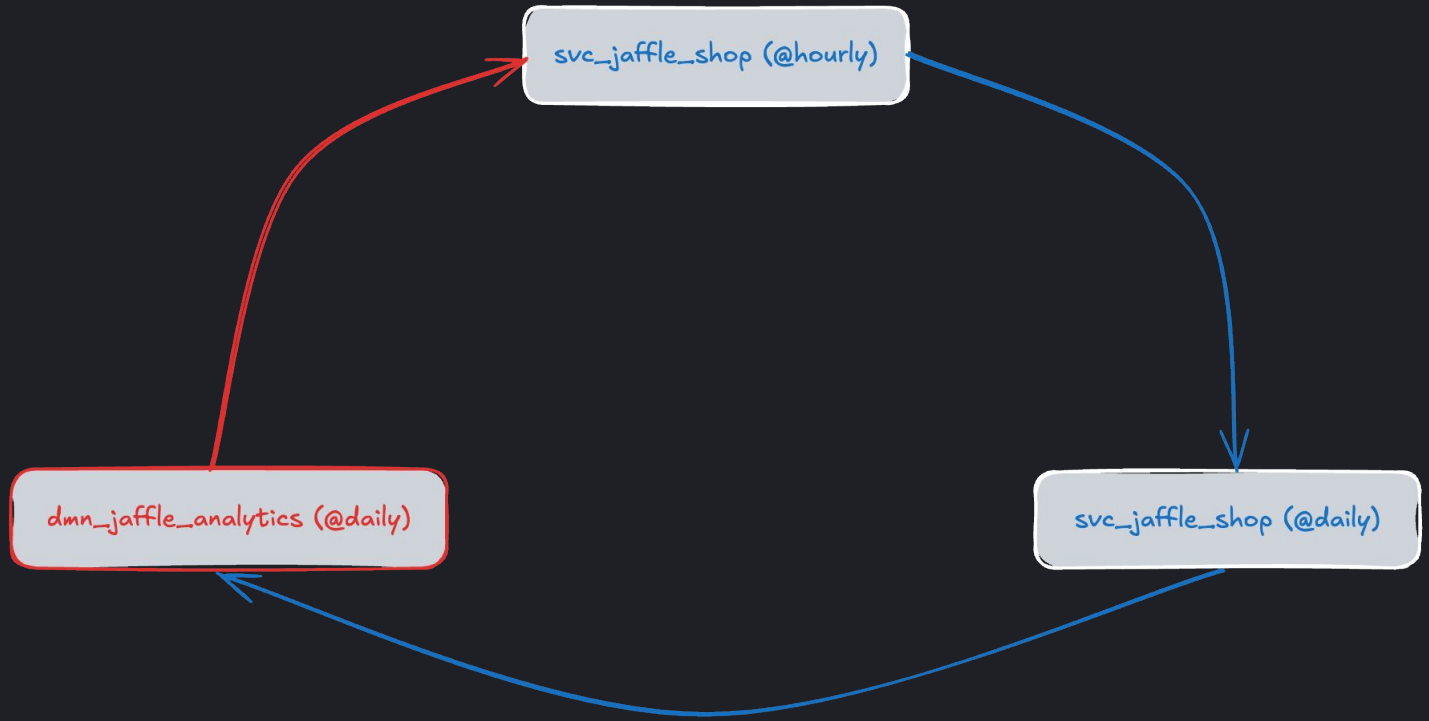
Analysts

Researchers

Data Engineers

Боль №0: на сколько сильно дробить предметную область на домены?

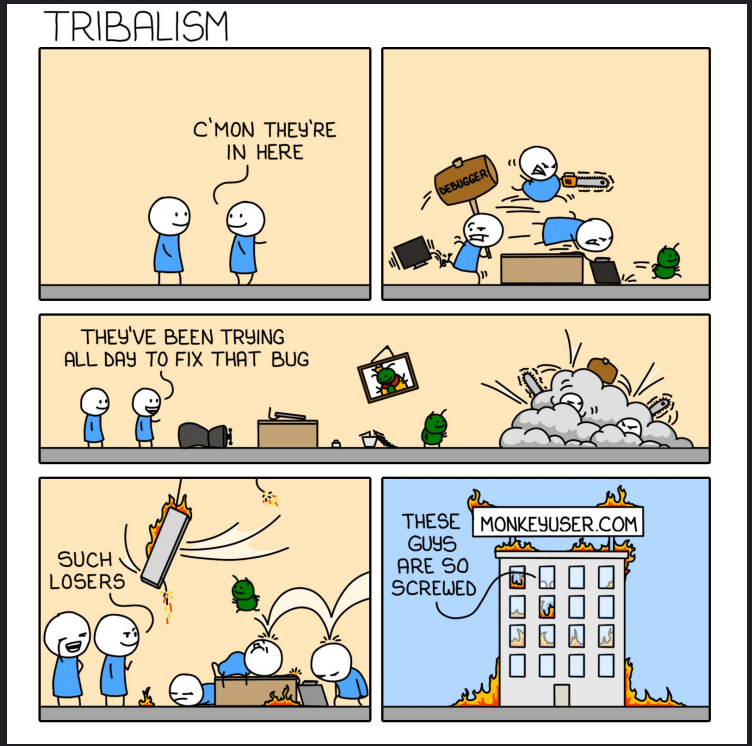
Боль №1: циклические зависимости



Боль №2: проблемные домены

https://www.monkeyuser.com/

Боль №3: восстановление после инцидентов



https://www.monkeyuser.com/

Плюсы Data Mesh

- Масштабирование: возможность управлять большим хранилищем силами небольшой команды
- Velocity/time to market
- Низкий порог входа для пользователей дата платформы

Минусы Data Mesh

- Data Mesh требует от пользователей понимания принципов архитектуры
- Data Mesh приносит дополнительную техническую сложность в инфраструктуру

Выводы

- Data Mesh топ, обратно на монолит не хотим
- Инфра местами топ, но не совсем
- Инструменты местами не топ

@lkozhinov

@nikitayurasov

dbt-af

Toloka