Bestplace

Оркестратор пайплайнов для небольшой команды инженеров и аналитиков: как мы используем dagster

Дмитрий Крылов
mityakrylov

Алексей Завальский
al_zav

Bestplace

Геоаналитическая платформа

- Геоданные (координаты, геометрии)
- Аналитика (визуализация, обработка)
- Машинное обучение

Компания

- Основана в 2016 году
- Является одним из лидеров рынка геоаналитики в России

Клиенты и партнеры

- Ритейл
- FMCG
- Сервисы поиска недвижимости

Контекст: данные

Различные источники:

- парсинг веба
- государственные открытые данные
- партнерские и платные api
- OpenStreetMap

Ну и что ты тут накачал?

Контекст: данные

Различные источники:

- парсинг веба
- государственные открытые данные
- партнерские и платные api
- OpenStreetMap

Храним в mongodb и S3

Ну и что ты тут накачал?

Контекст: данные

Различные источники:

- парсинг веба
- государственные открытые данные
- партнерские и платные api
- OpenStreetMap

Храним в mongodb и S3

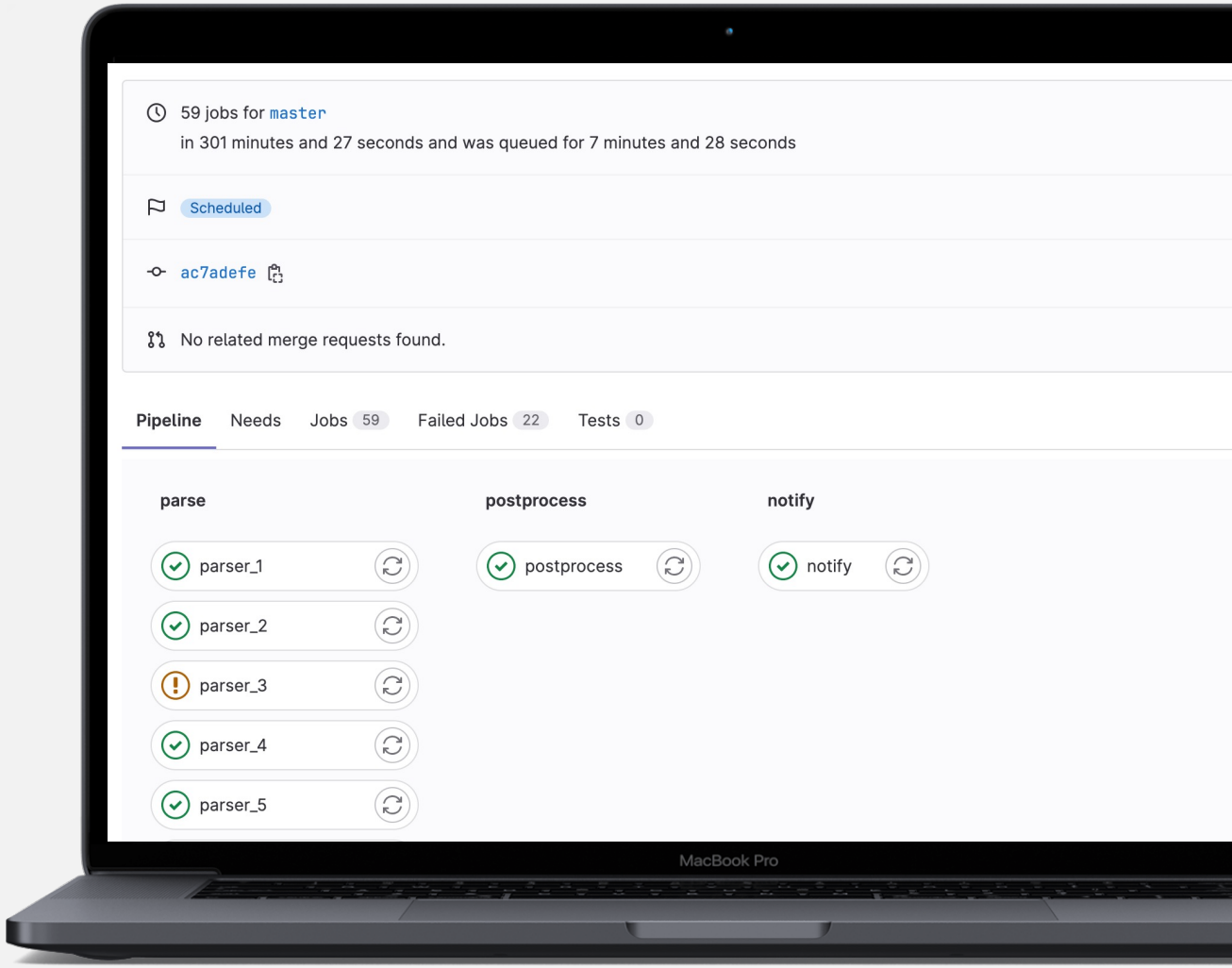
Данных много разных по смыслу (домохозяйства, точки интереса, трафики и т.д.), но немного по объему (десятки Тб)

Ну и что ты тут накачал?

Контекст: пайплайны

Часть пайплайнов писали разработчики в gitlab ci:

- запуск по расписанию
- простые графы зависимостей

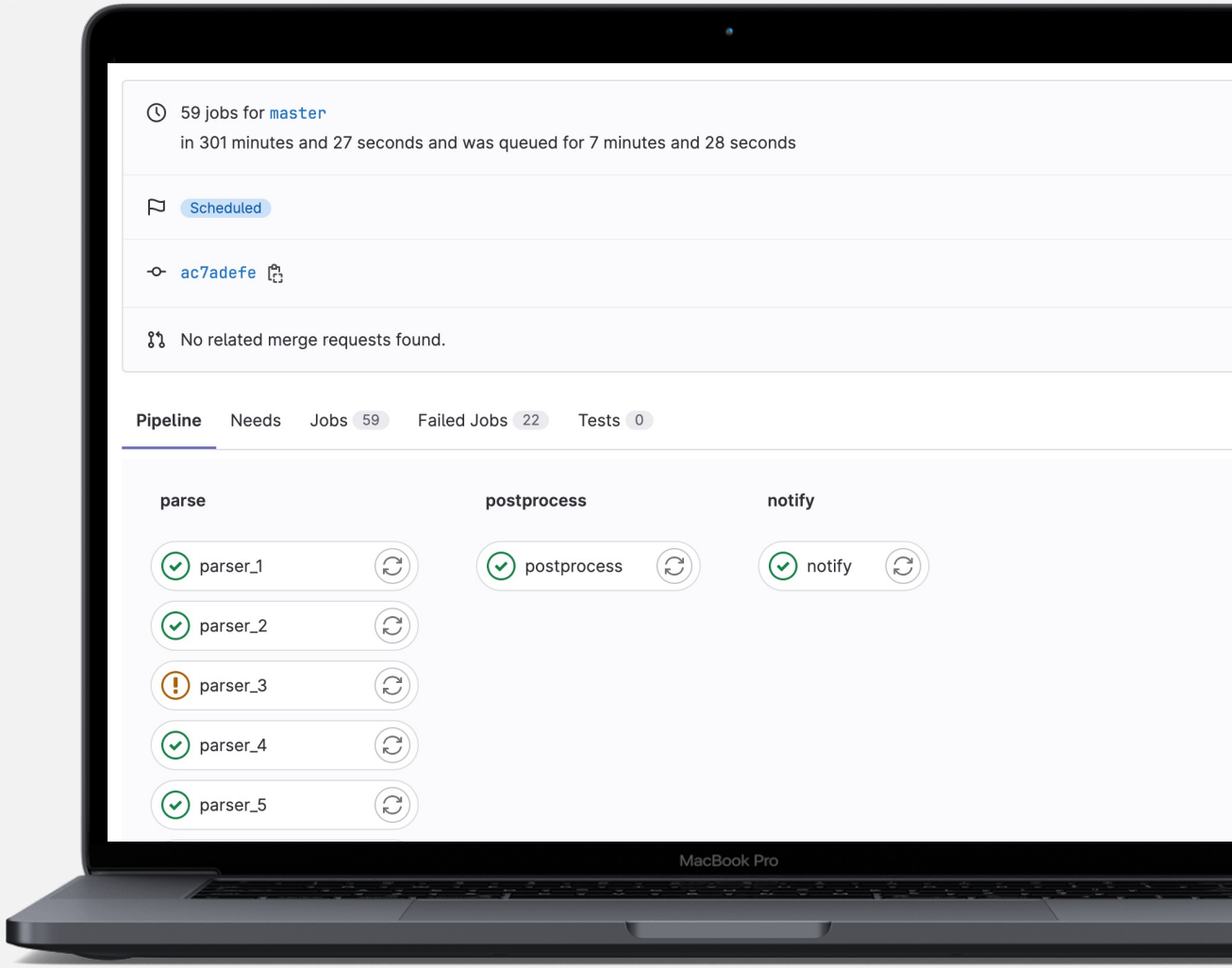


Контекст: пайплайны

Часть пайплайнов писали разработчики в gitlab ci:

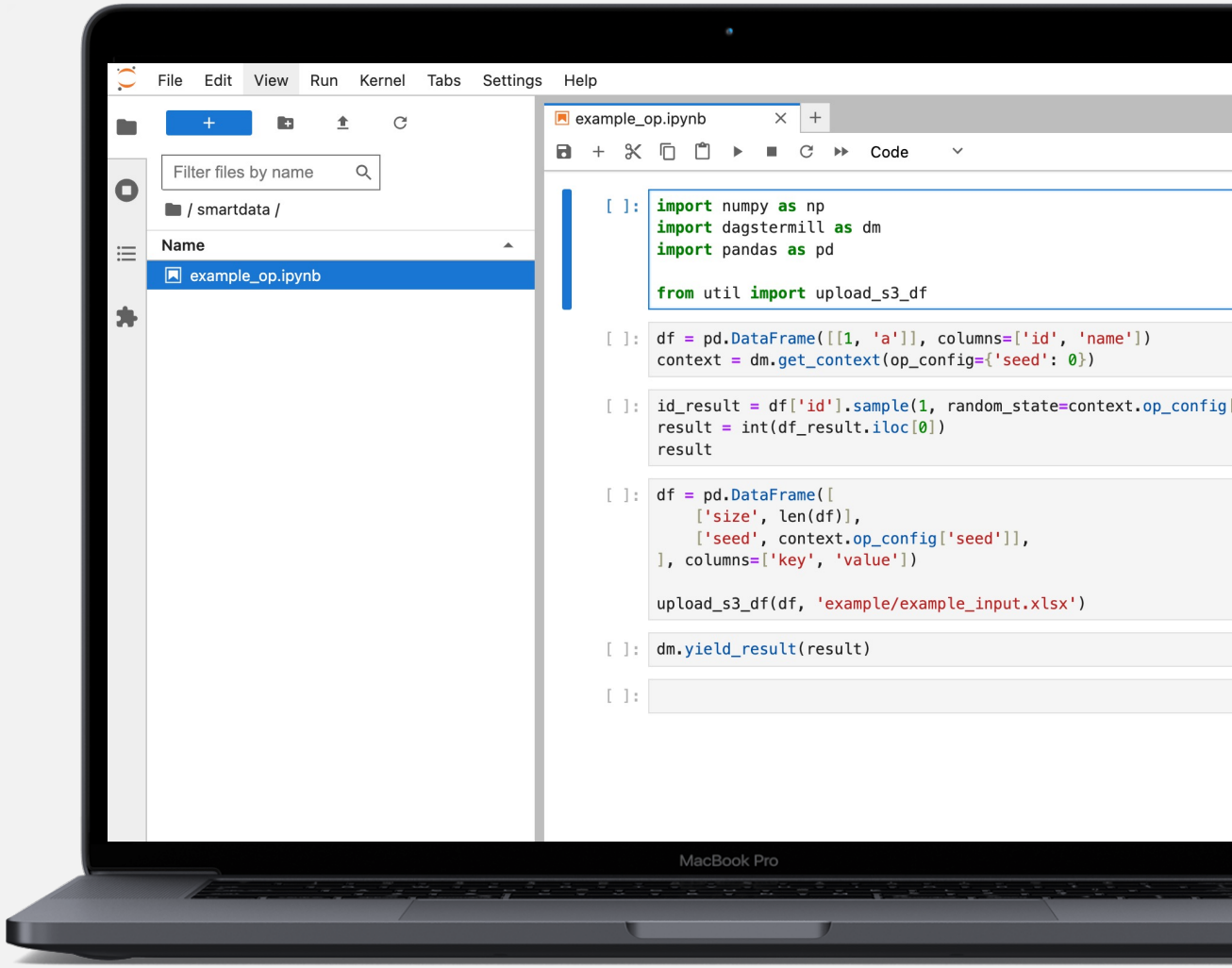
- запуск по расписанию
- простые графы зависимостей

Проблемы: неудобно следить и параметризировать



Контекст: пайплайны

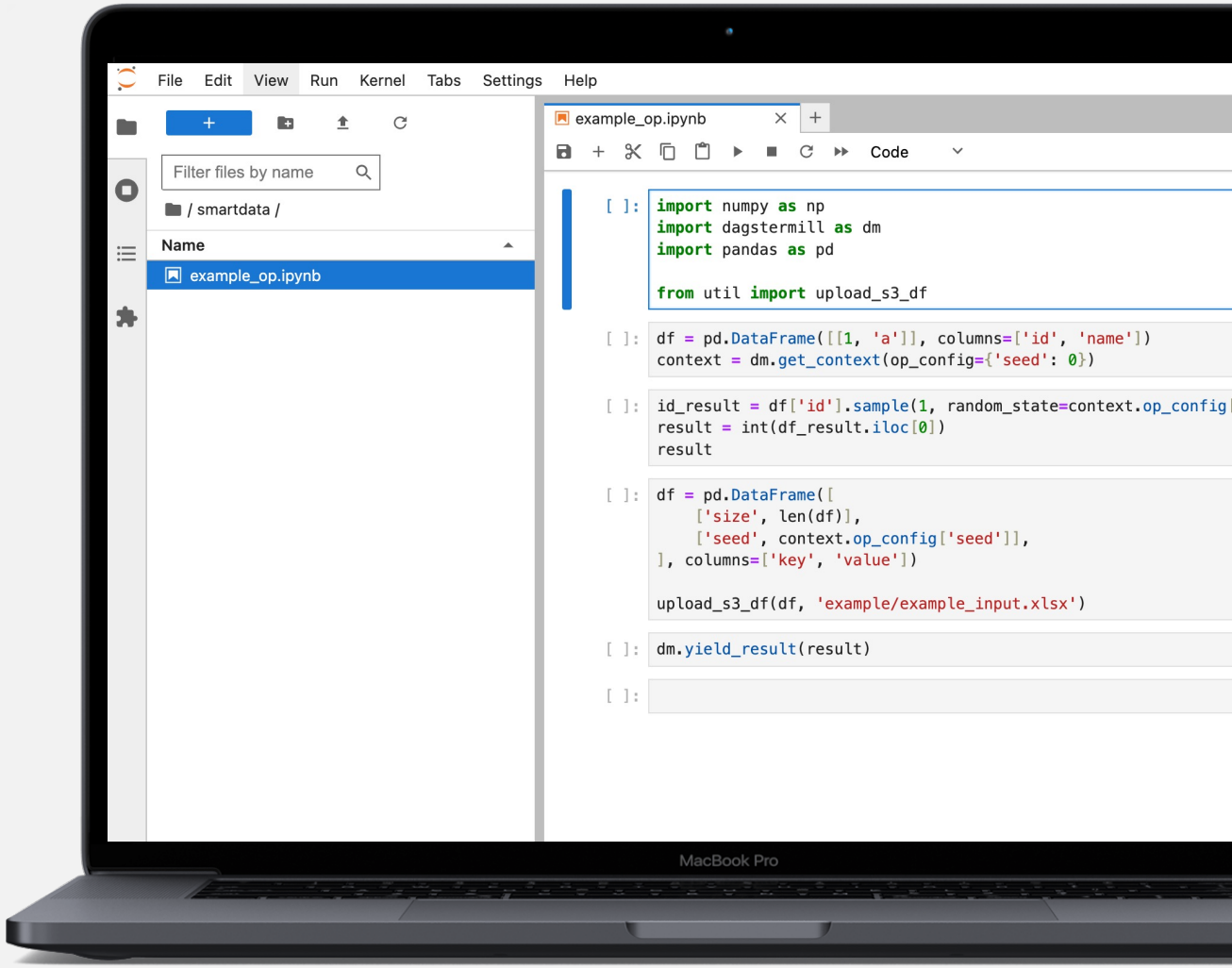
Часть пайплайнов писали аналитики и дата саентисты в jupyter ноутбуках



Контекст: пайплайны

Часть пайплайнов писали аналитики и дата саентисты в jupyter ноутбуках

Проблемы: неудобно воспроизводить



Выбор инструмента, 2020 год

Пожелания:

- Несложно разобраться и развернуть
- Параметризируемый запуск из интерфейса
- Интеграция с jupyter
- Симпатичный UI

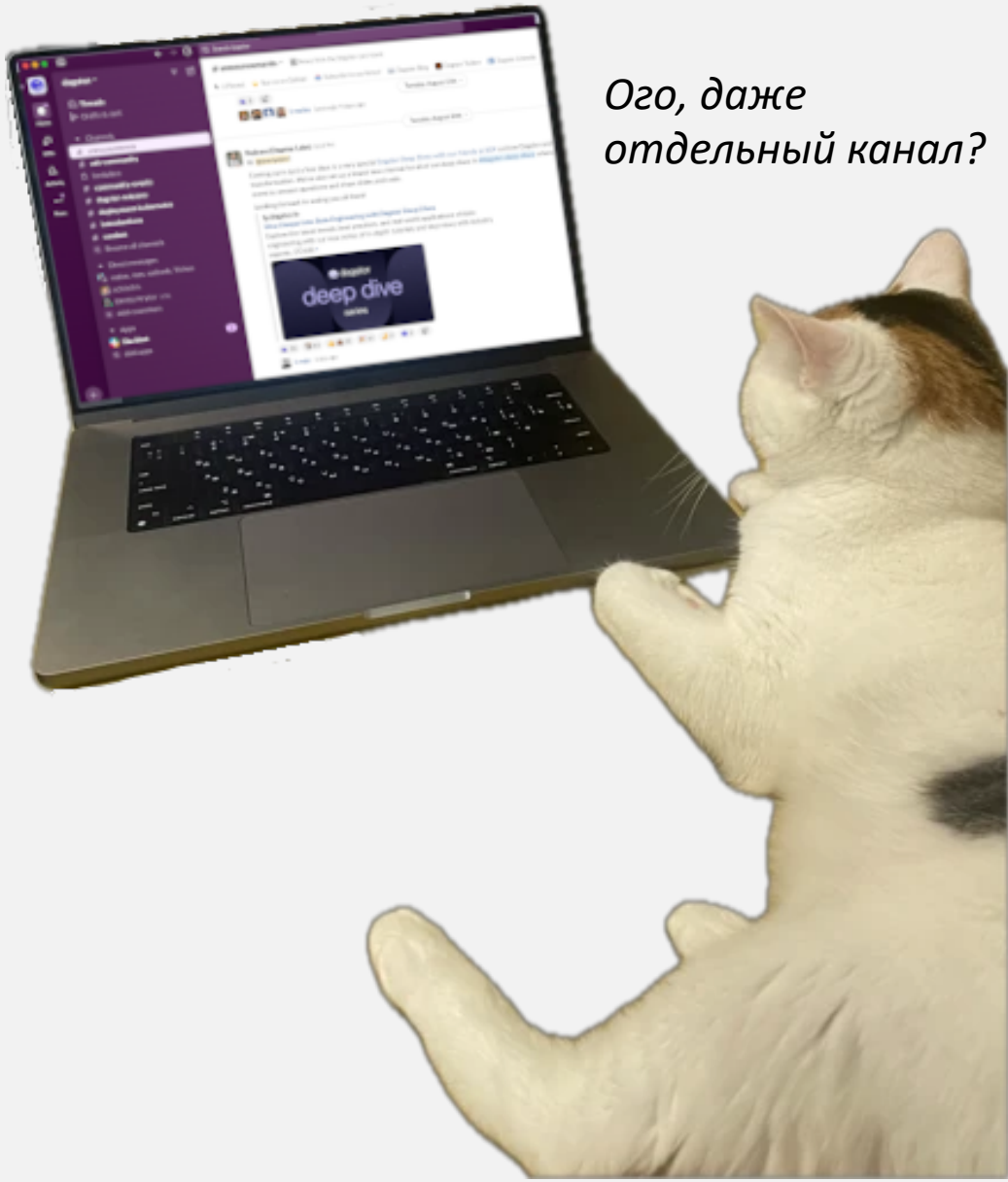
Dagster

Что понравилось:

- Соответствовал всем пожеланиям
- Хорошая документация
- Скорость обновлений

Коммуникация с командой dagster

github, slack, созвоны



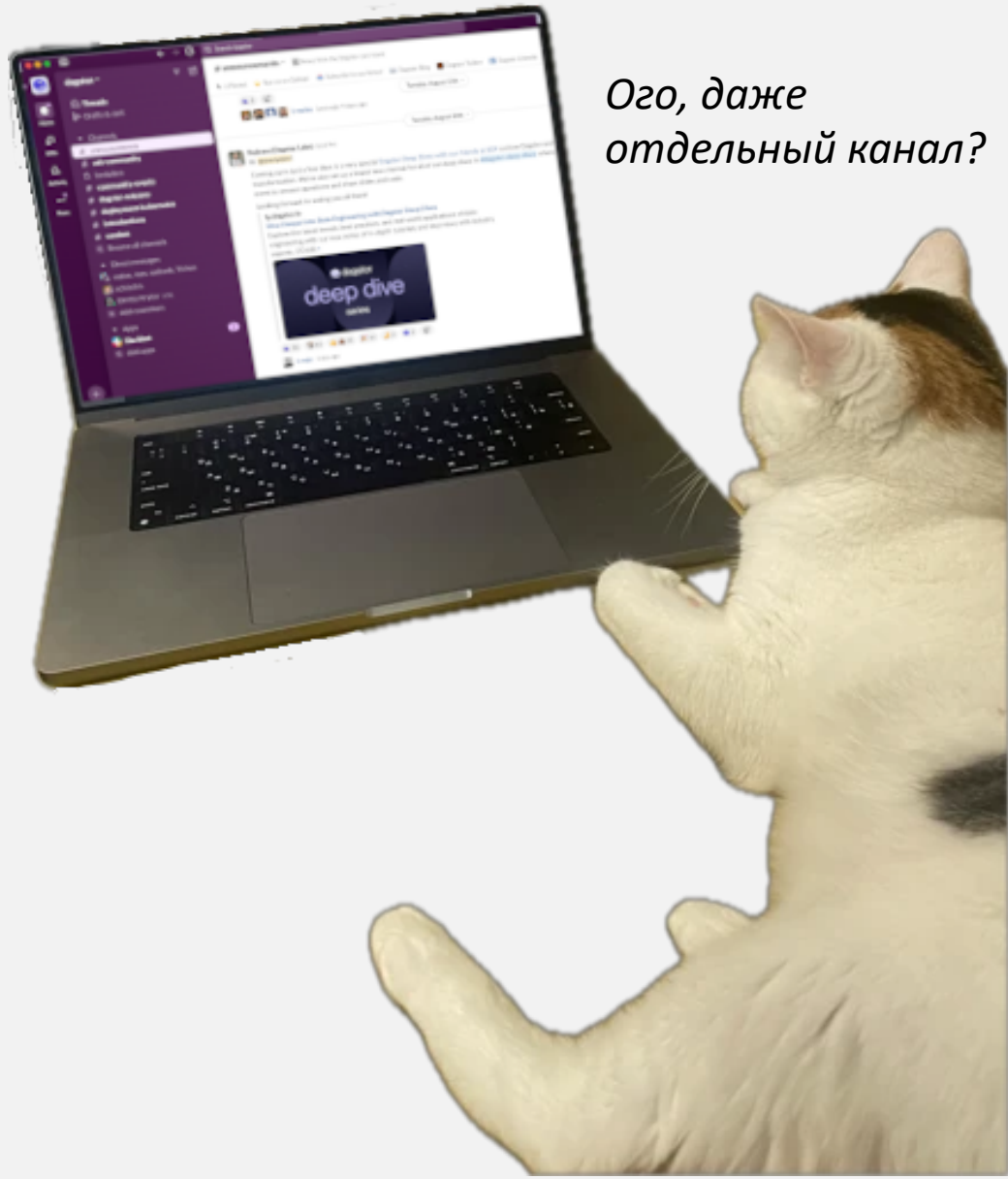
Ого, даже отдельный канал?

Коммуникация с командой dagster

github, slack, созвоны

Совместные улучшения:

- docker in docker executor
- фикс взаимодействия с minio
- фиксы взаимодействия с jupyter
- и другие



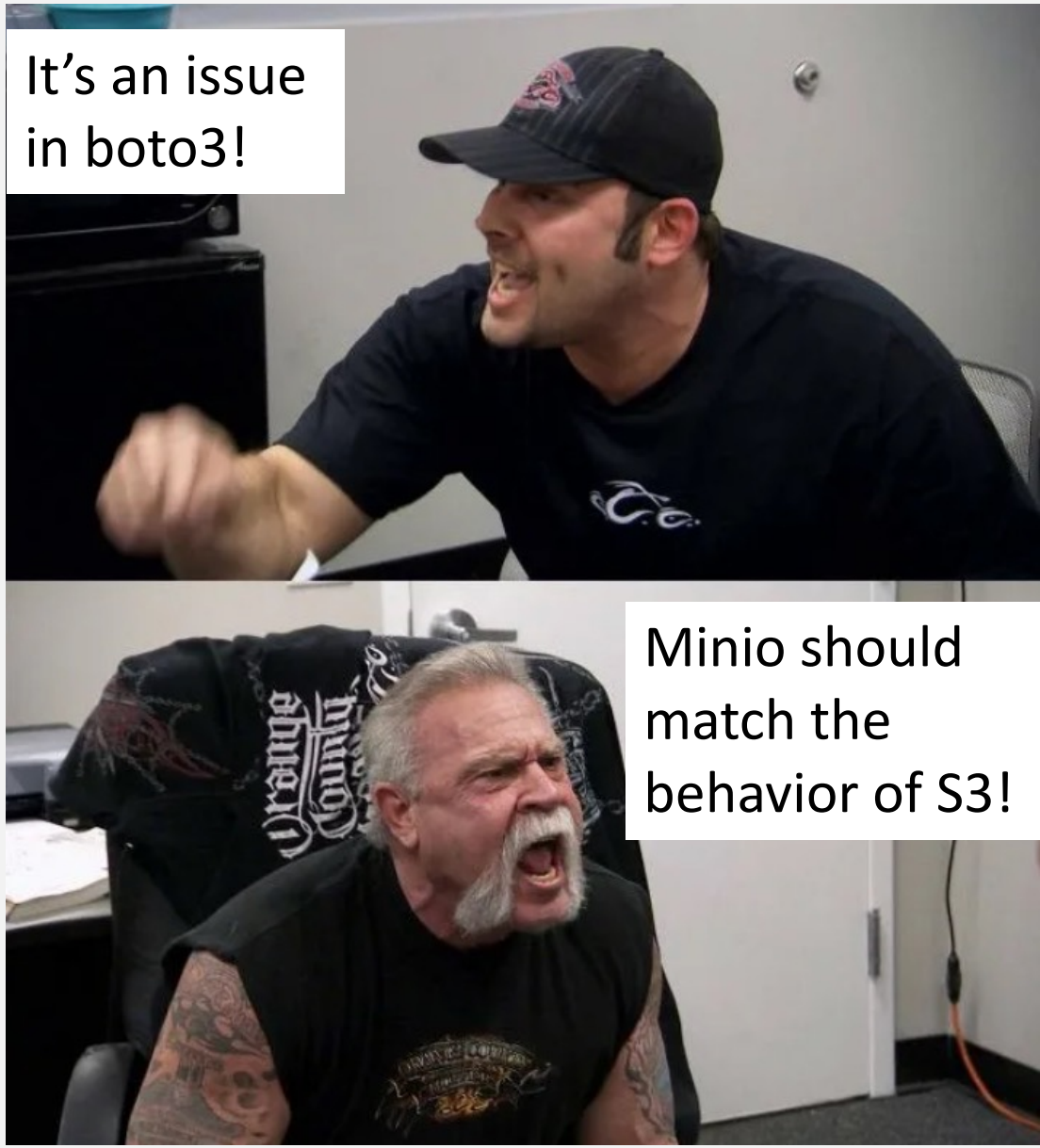
Ого, даже отдельный канал?

Коммуникация с командой dagster

github, slack, созвоны

Совместные улучшения:

- docker in docker executor
- фикс взаимодействия с minio
- фиксы взаимодействия с jupyter
- и другие

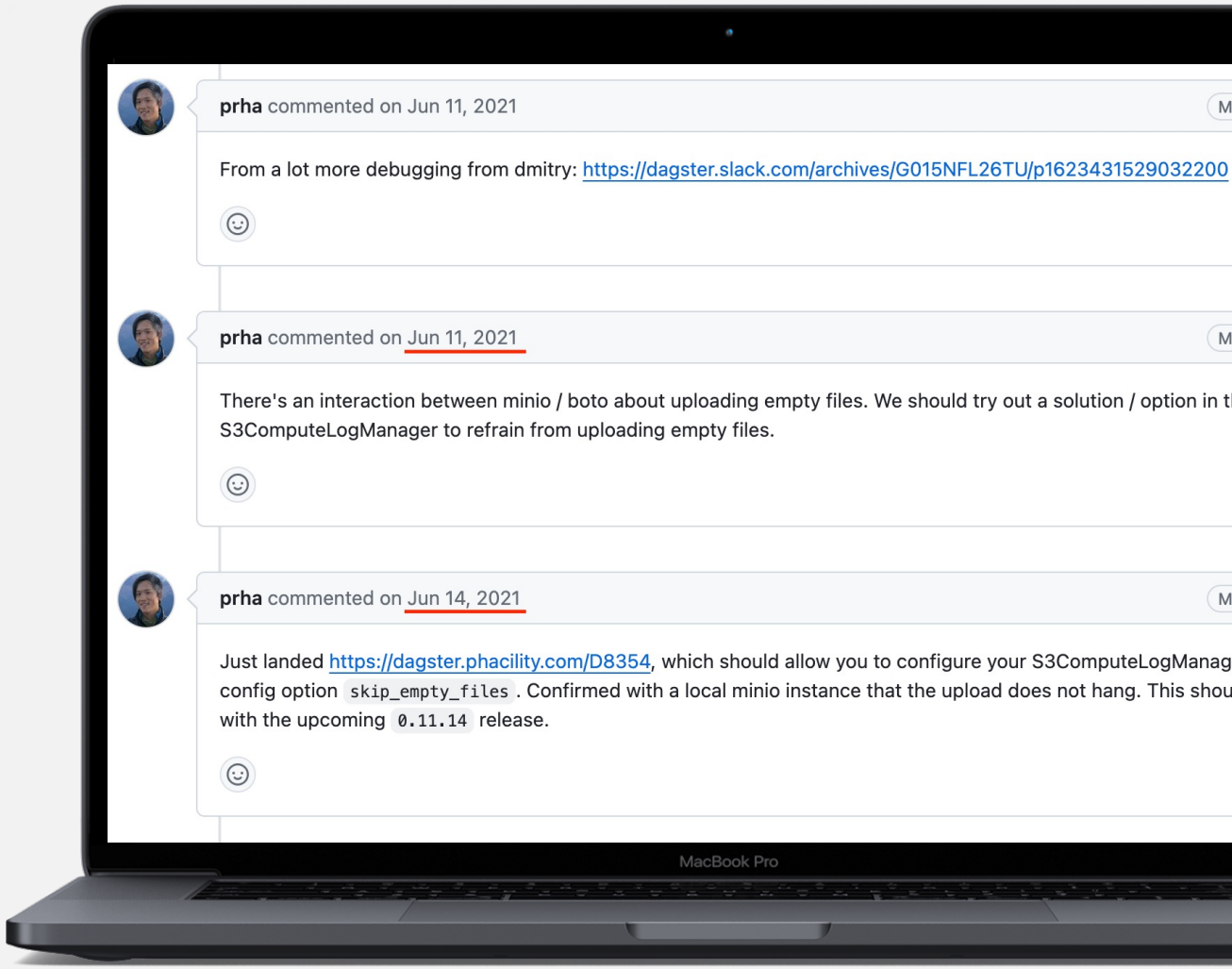


Коммуникация с командой dagster

github, slack, созвоны

Совместные улучшения:

- docker in docker executor
- фикс взаимодействия с minio
- фиксы взаимодействия с jupyter
- и другие



Коммуникация с командой dagster

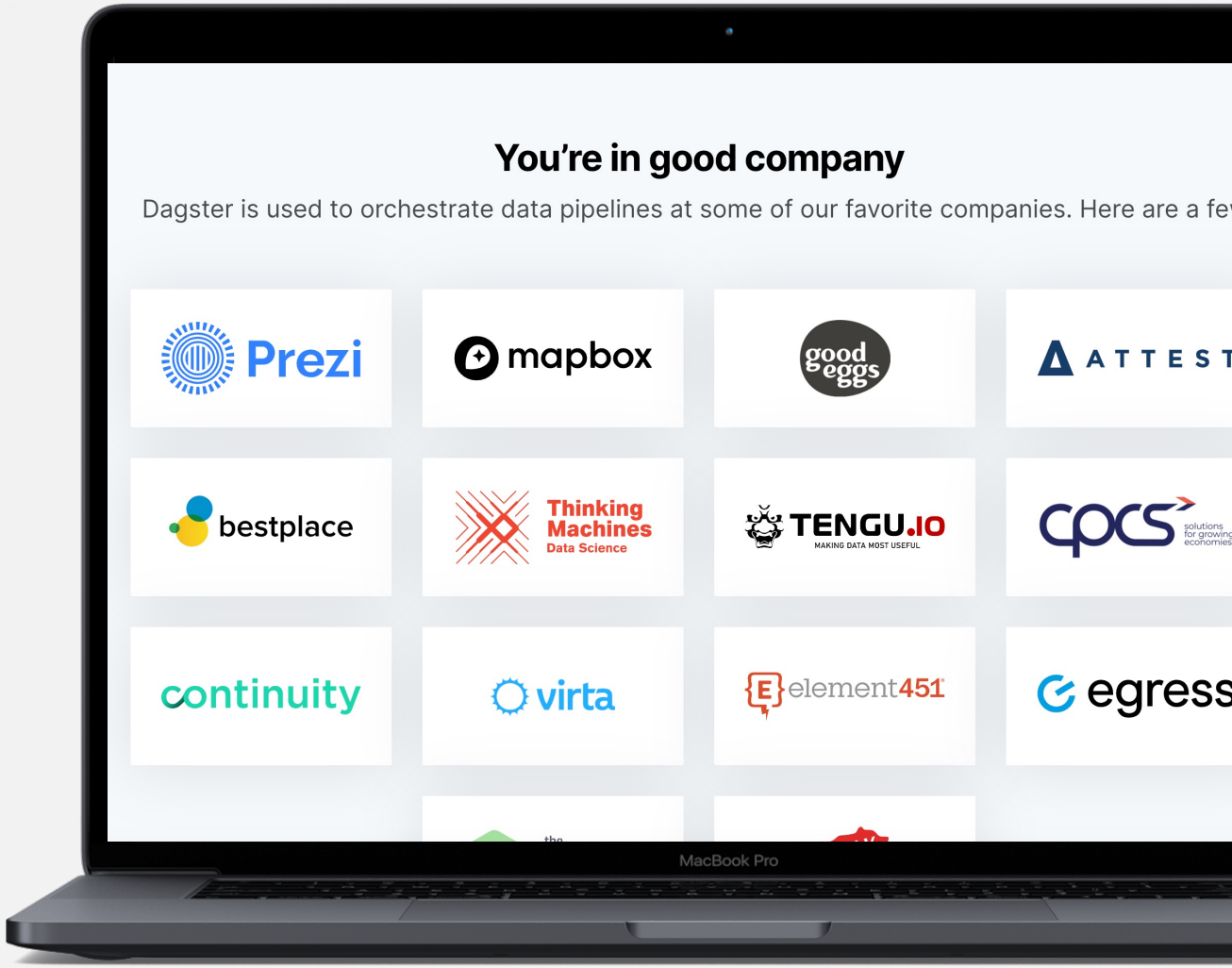
github, slack, созвоны

Совместные улучшения:

- docker in docker executor
- фикс взаимодействия с minio
- фиксы взаимодействия с jupyter
- и другие

Релизы в течение недели после исправлений.

Взаимодействуйте с open source командами!



Коммуникация с командой dagster

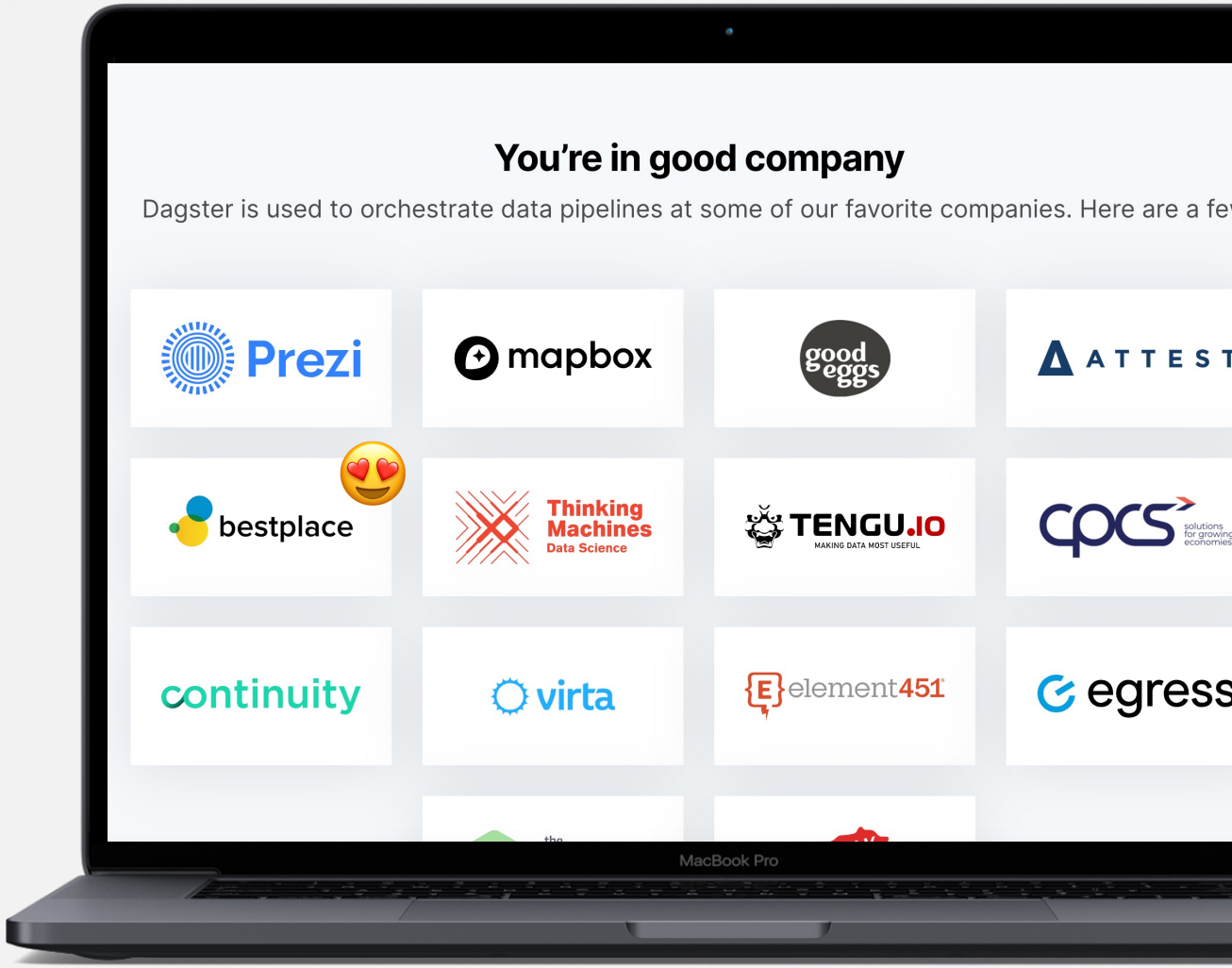
github, slack, созвоны

Совместные улучшения:

- docker in docker executor
- фикс взаимодействия с minio
- фиксы взаимодействия с jupyter
- и другие

Релизы в течение недели после исправлений.

Взаимодействуйте с open source командами!



Результат внедрения

Результат для дата-инженеров

Фреймворк для создания пайплайнов:

- op
- graph
- job
- schedule

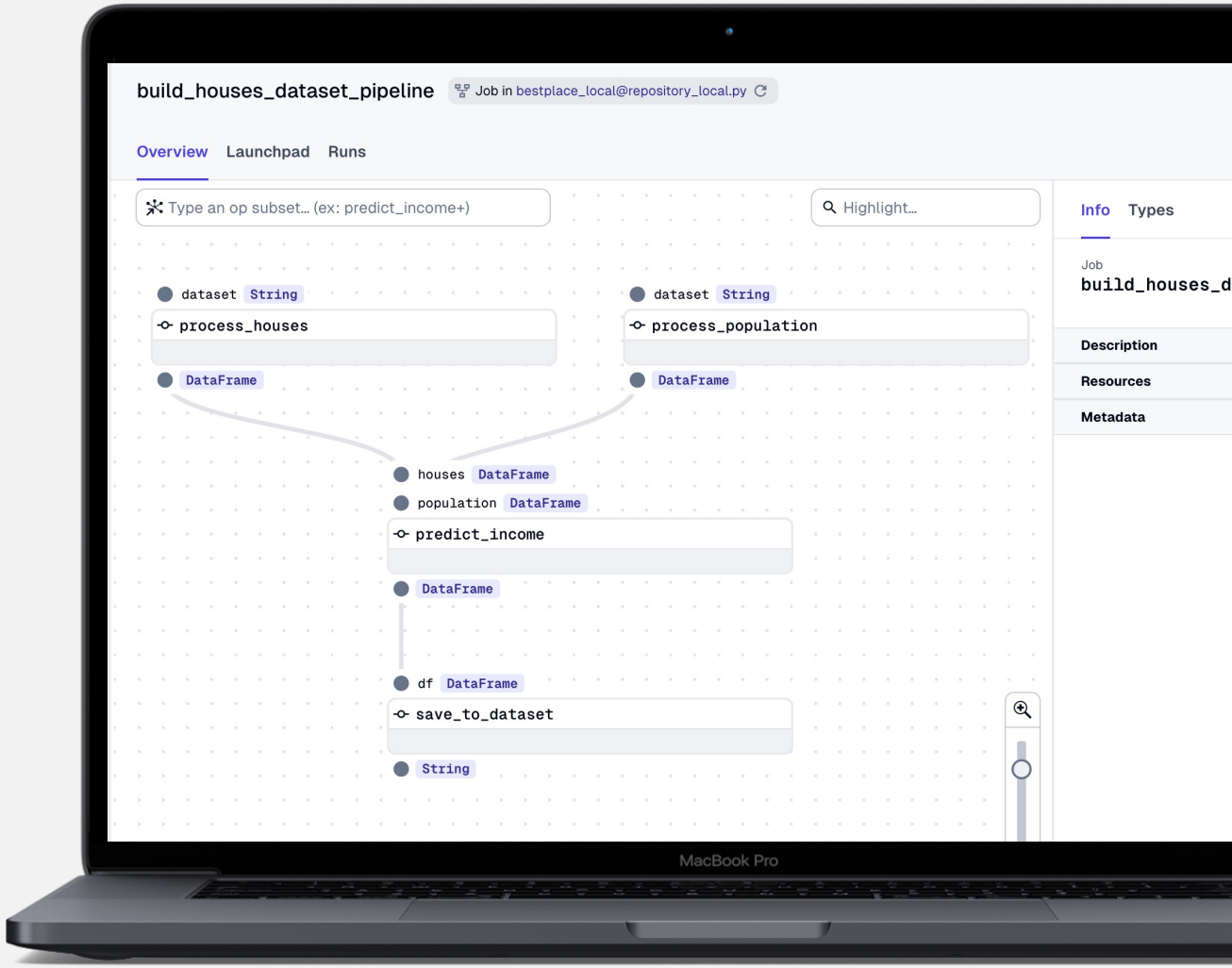
```python
import pandas as pd
from dagster import op, graph

@op
def process_houses(dataset: str) -> pd.DataFrame:...

@op
def process_population(dataset: str) -> pd.DataFrame:...

@op
def predict_income(houses: pd.DataFrame, population: pd.DataFrame) -> pd.DataFrame:..

@op
def save_to_dataset(df: pd.DataFrame) -> str:...


@graph
def build_houses_dataset_pipeline():
    dataset = predict_income(process_houses(), process_population())
    save_to_dataset(dataset)
```

Результат для дата-инженеров

Фреймворк для создания пайплайнов:

- op
- graph
- job
- schedule

Удобное отслеживание статуса и логов запусков



Результат для аналитиков

Пайплайны из jupyter ноутбуков
с помощью dagstermill (papermill)

```python
import pandas as pd
import dagstermill as dm
from dagster import In, Out, Any


process_dataset_op = dm.define_dagstermill_op(
    name='process_dataset',
    notebook_path='notebooks/process_dataset_op.ipynb',
    ins={
        'df': In(pd.DataFrame, description='Input dataframe')
    },
    outs={
        'result': Out(Any)
    },
)
```

Результат для аналитиков

Пайплайны из jupyter ноутбуков
с помощью dagstermill (papermill)

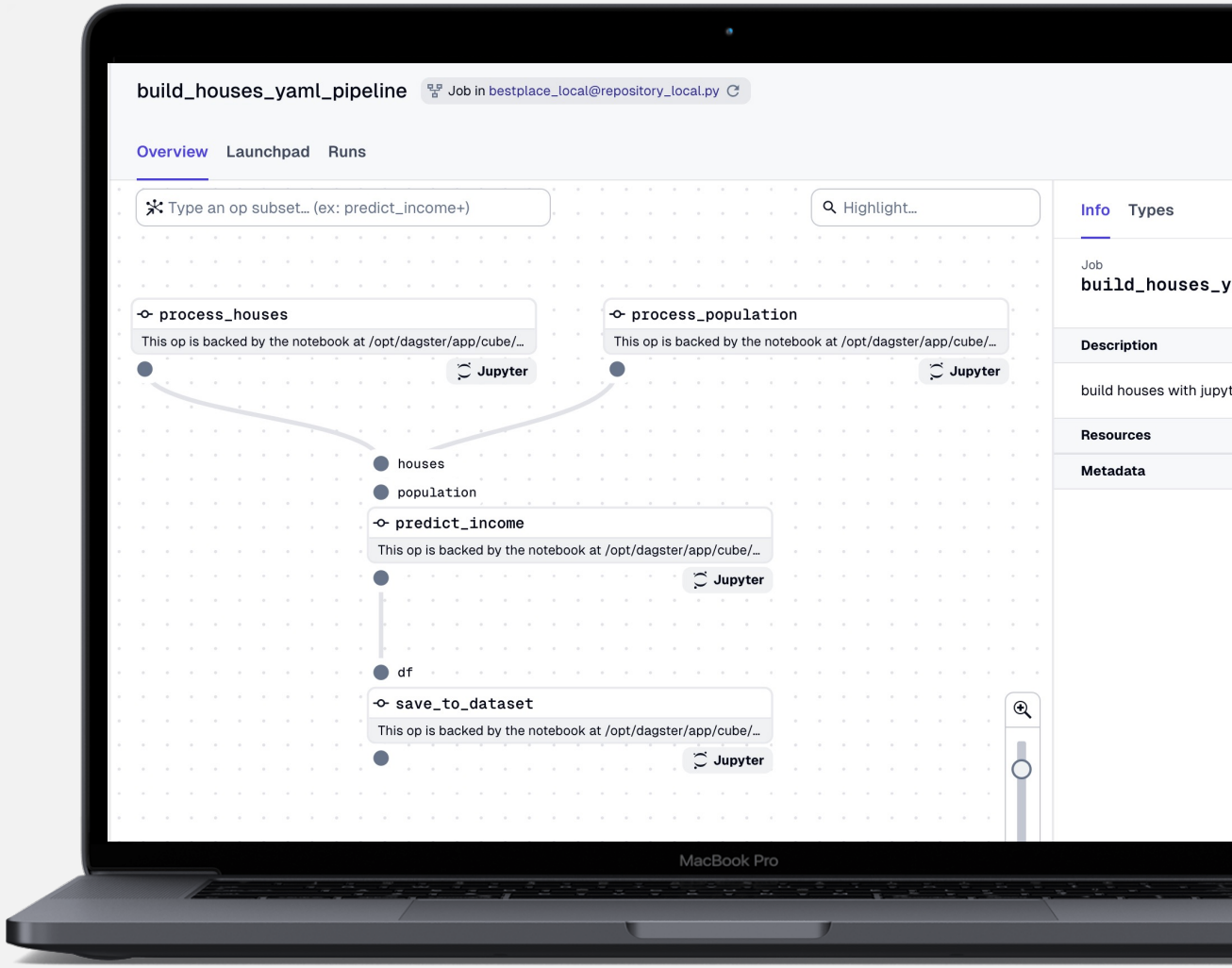
Надстройка над синтаксисом dagster:
yaml пайплайны

```yaml
name: build_houses_yaml_pipeline
description: build houses with jupyter notebooks as ops
ops:
  - def: build_houses/process_houses.ipynb
    alias: process_houses

  - def: build_houses/process_population.ipynb
    alias: process_population

  - def: build_houses/predict_income.ipynb
    alias: predict_income
    deps:
      houses: process_houses
      population: process_population

  - def: build_houses/save_to_dataset.ipynb
    alias: save_to_dataset
    deps:
      df: predict_income

```

Результат для аналитиков

Пайплайны из jupyter ноутбуков
с помощью dagstermill (papermill)

Надстройка над синтаксисом dagster:
yaml пайплайны



Результат для аналитиков

Пайплайны из jupyter ноутбуков
с помощью dagstermill (papermill)

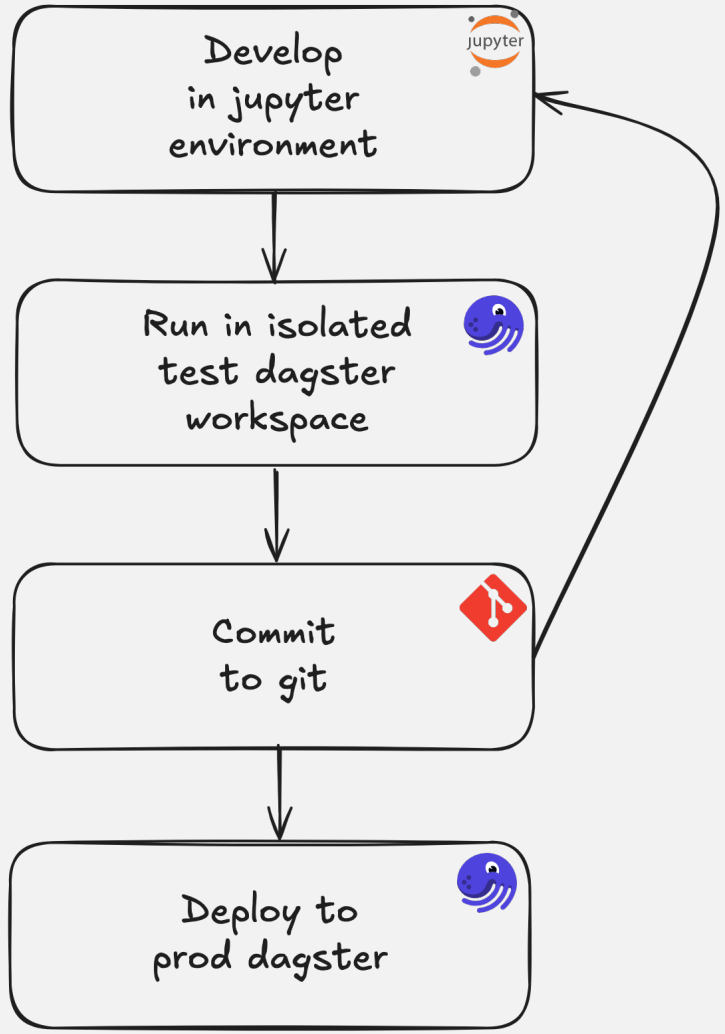
Надстройка над синтаксисом dagster:
yaml пайплайны

Результат для аналитиков

Пайплайны из jupyter ноутбуков
с помощью dagstermill (papermill)

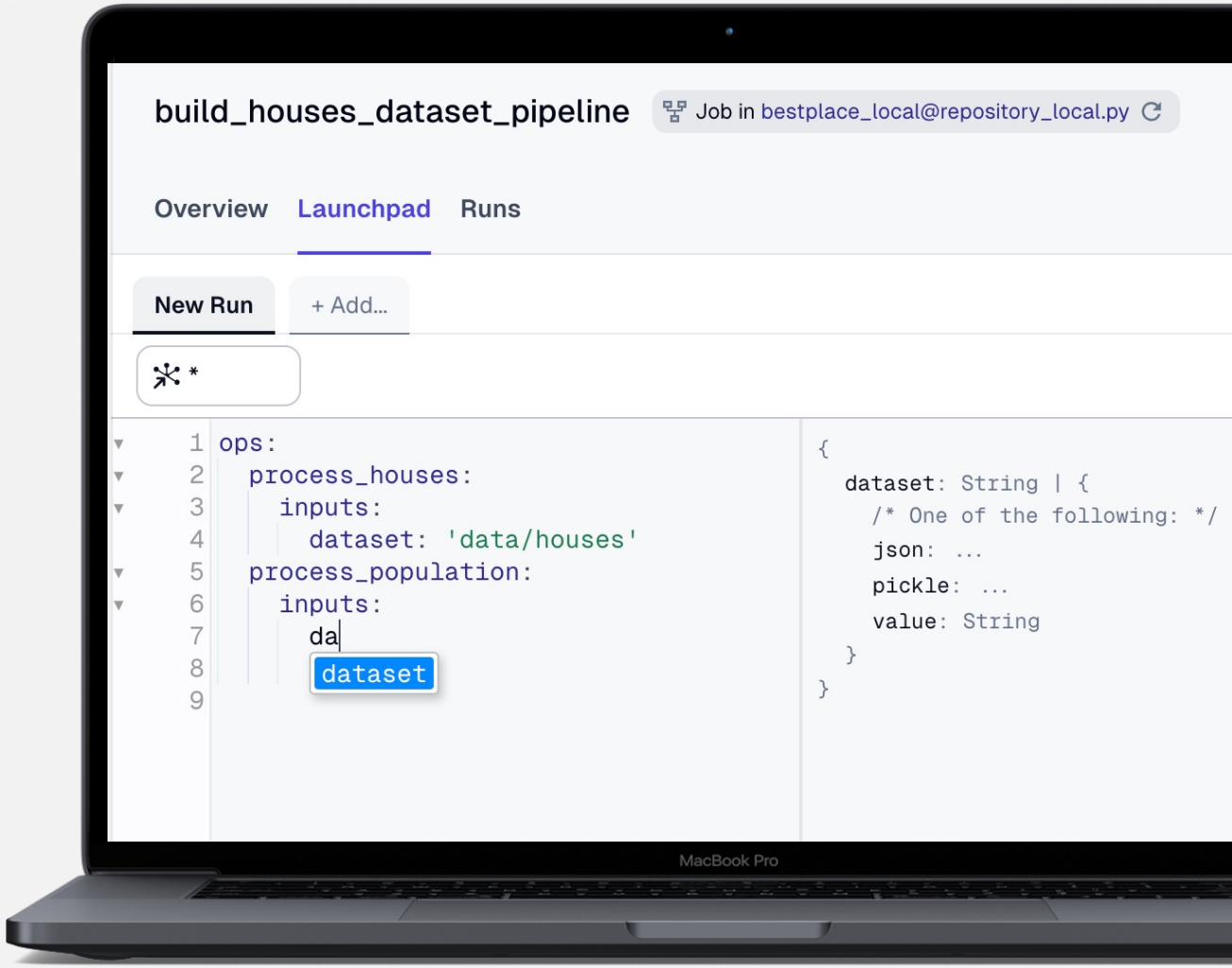
Надстройка над синтаксисом dagster:
yaml пайплайны

Тестовый контур



Результат для всех

Параметризируемый запуск по кнопке



Время историй

История про мониторинг запусков

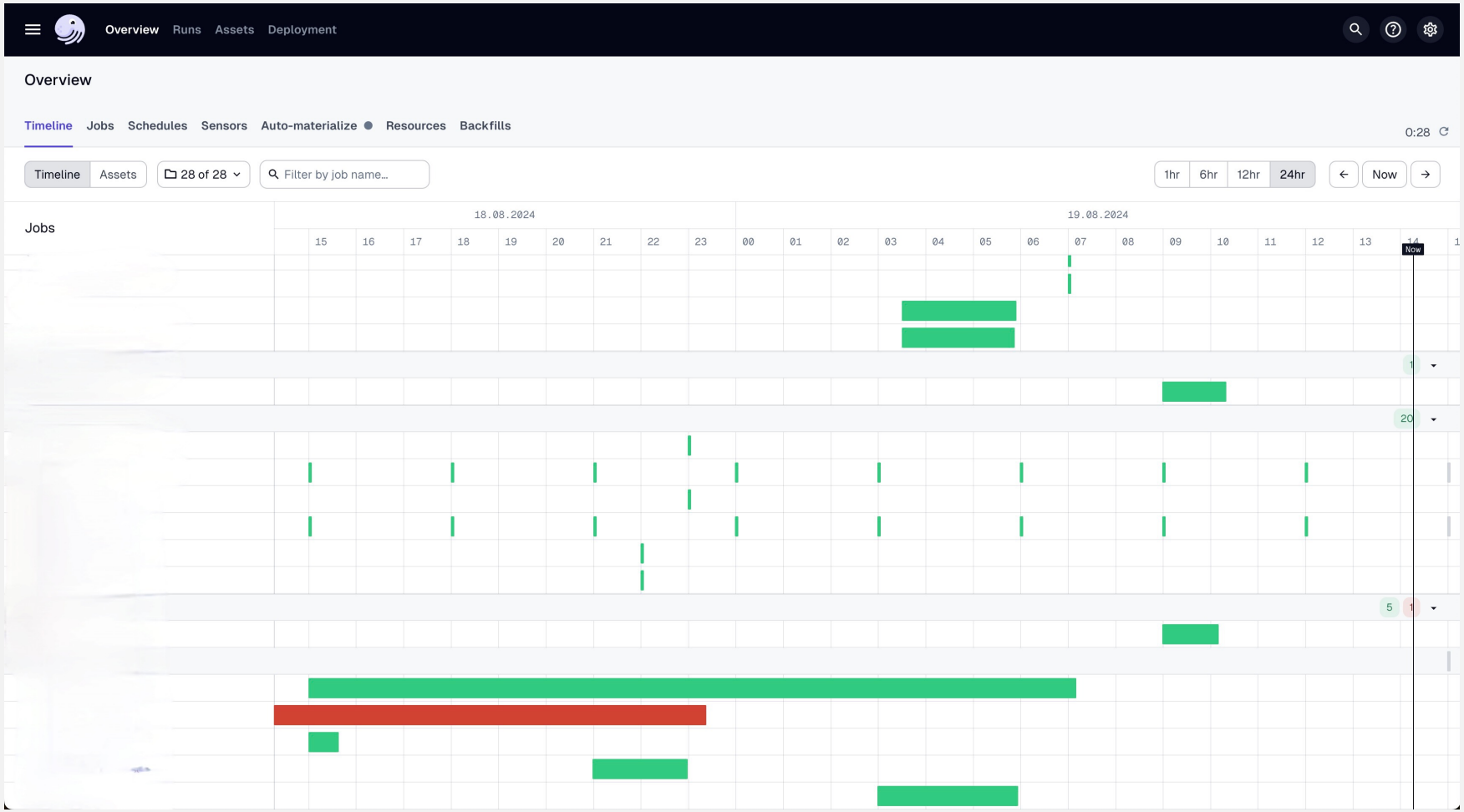
История про мониторинг запусков

Пользователи дагстера:

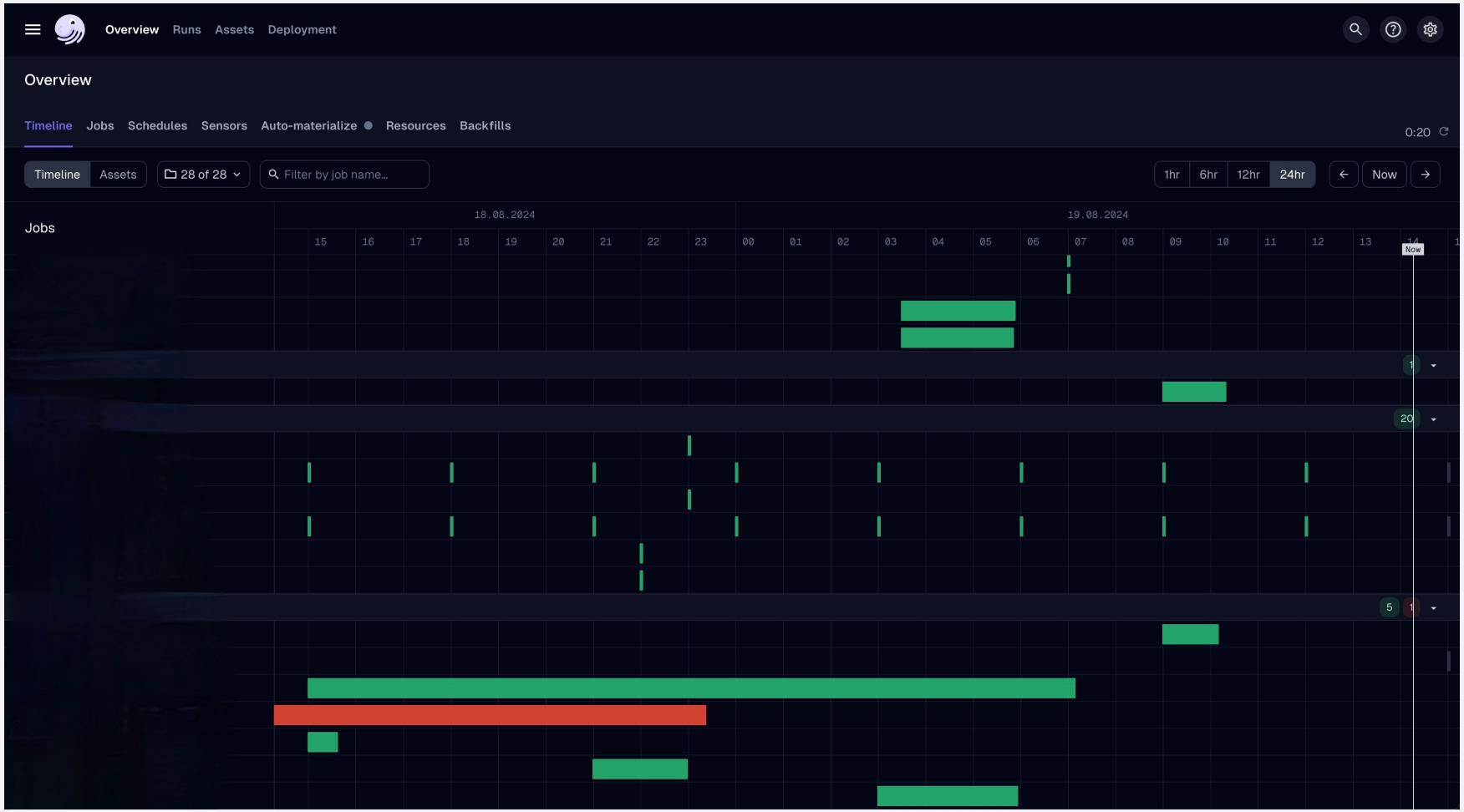
- дата-инженеры
- аналитики
- **менеджеры**

Не вижу,
куда смотреть

Мониторинг в UI



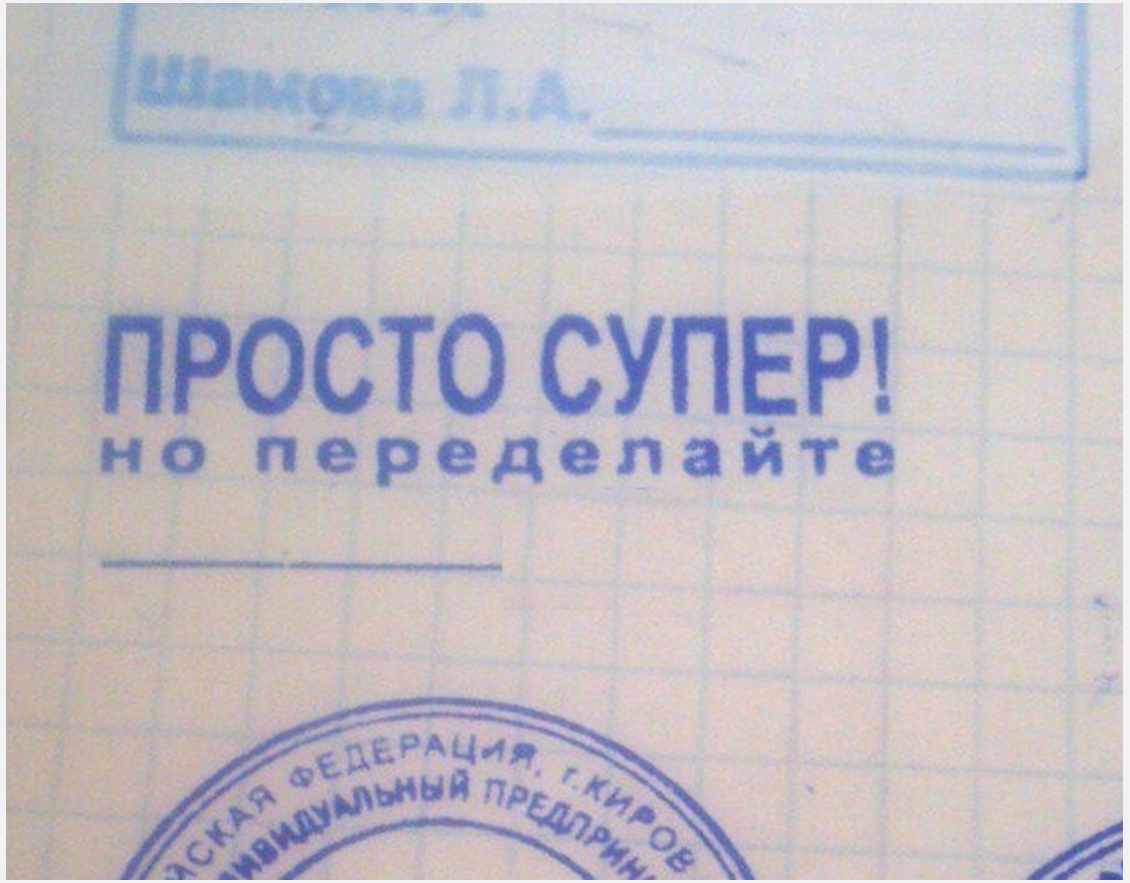
Мониторинг в UI



Мониторинг в telegram

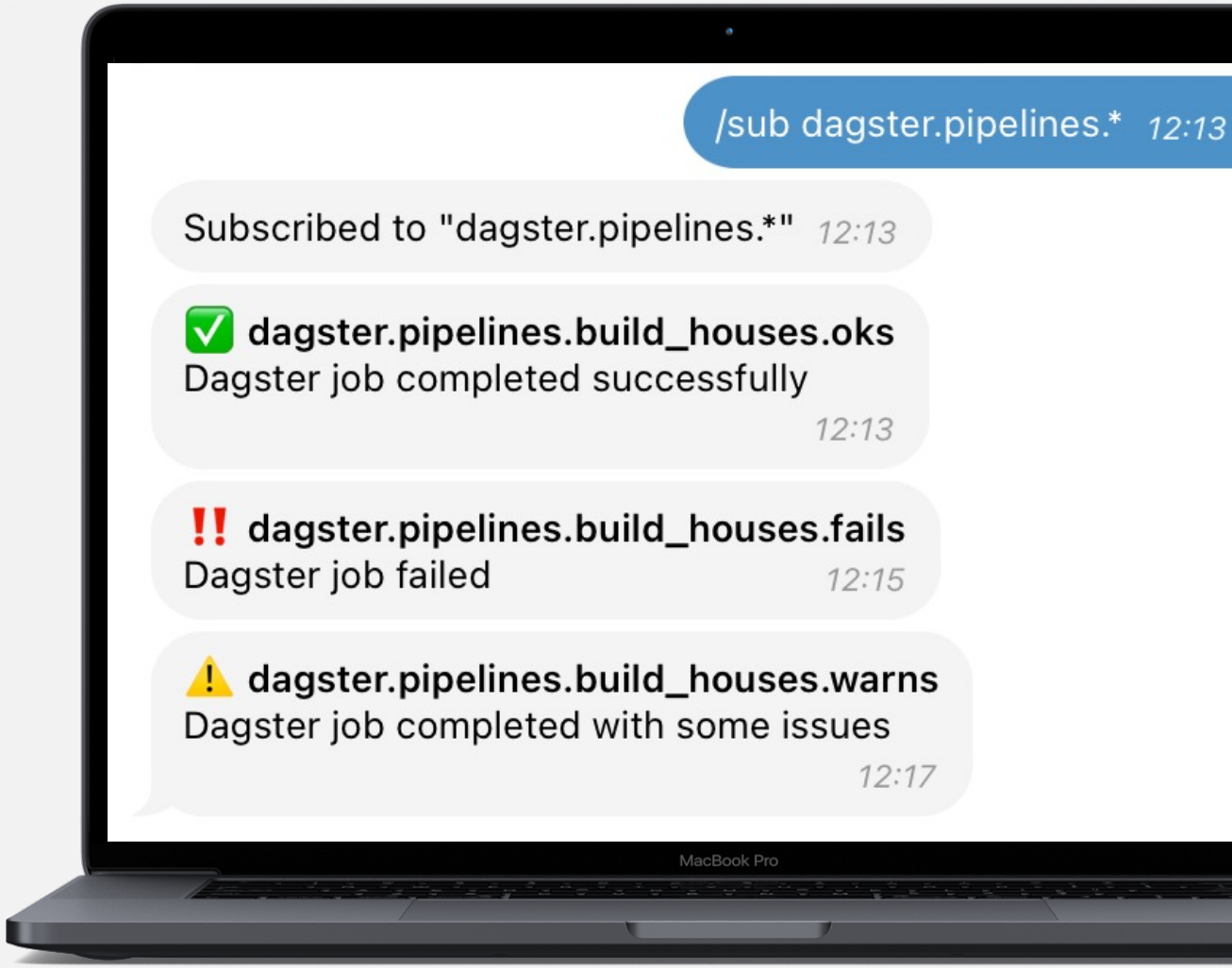
UI — это хорошо

Но хочется в телеграм



Мониторинг в telegram

TAAS (telegram as a service) — publish-subscribe сервис для отправки уведомлений через телеграм

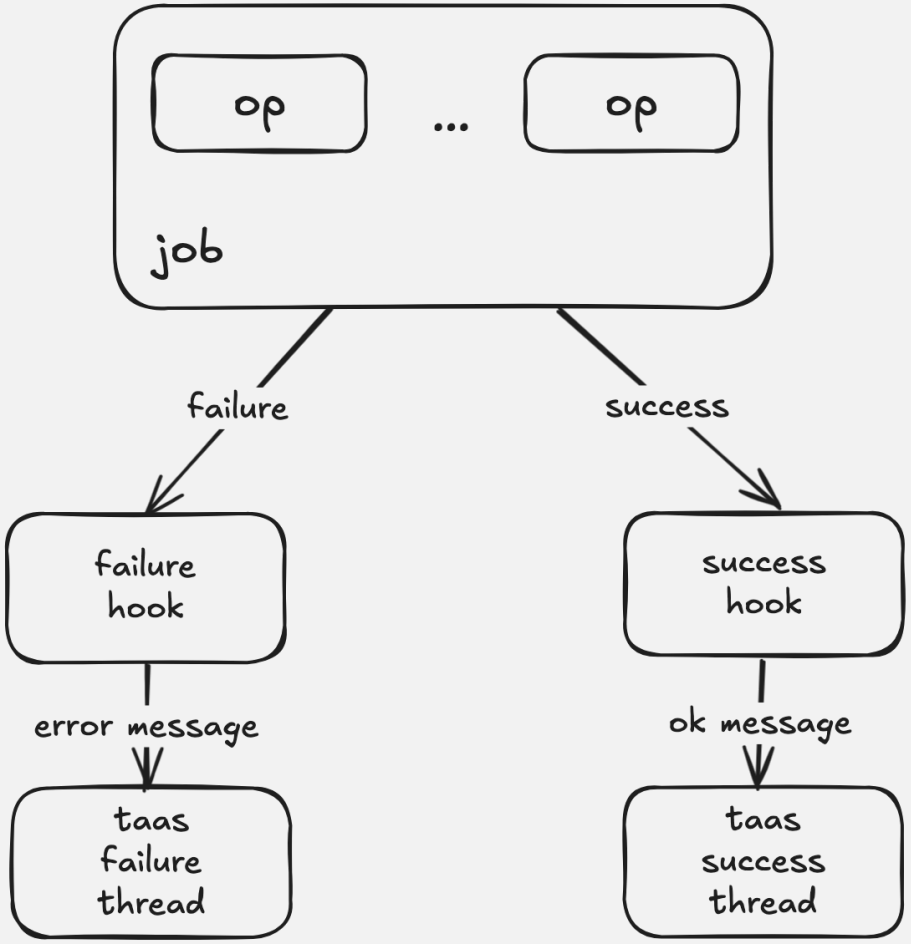


Мониторинг в telegram

TAAS (telegram as a service) — publish-subscribe сервис для отправки уведомлений через телеграм

Dagster hooks — механизм обработки завершившихся запусков

Мониторинг в telegram




```python
from dagster import HookContext, job, op, success_hook
from taas import submit_event


@success_hook
def taas_success_hook(context: HookContext):
    message = f'Dagster job {context.job_name} completed.'
    submit_event(f'bp.dagster.oks', message)


@op
def process_spatial_data():...


@job
def spatial_job():
    process_spatial_data.with_hooks({taas_success_hook})()
```

Итоги внедрения

Для менеджера и аналитика:

- уведомления в удобном формате

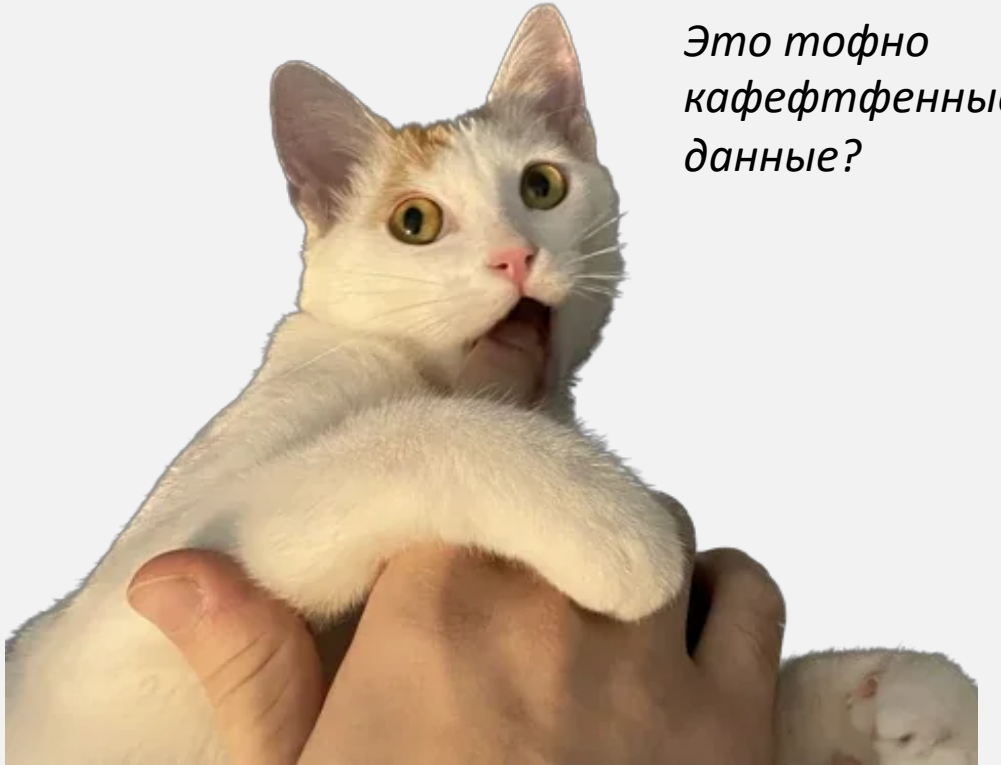
Для дата-инженера:

- простой способ добавлять уведомления

История про качество данных

История про качество данных

- Мониторинга недостаточно
- Пайплайны работают как надо, но есть нюанс
- Требуются проверки качества данных



Это тофно кафефтфенные данные?

Немного data quality

- great expectations
- **pandera**

Ваши expectations —
ваши проблемы!

Data quality: pandera

- Проще в использовании
- Изначально ориентирована на работу с pandas
- Схожесть с уже известным pydantic

Почему тебе пишет какая-то пантера?

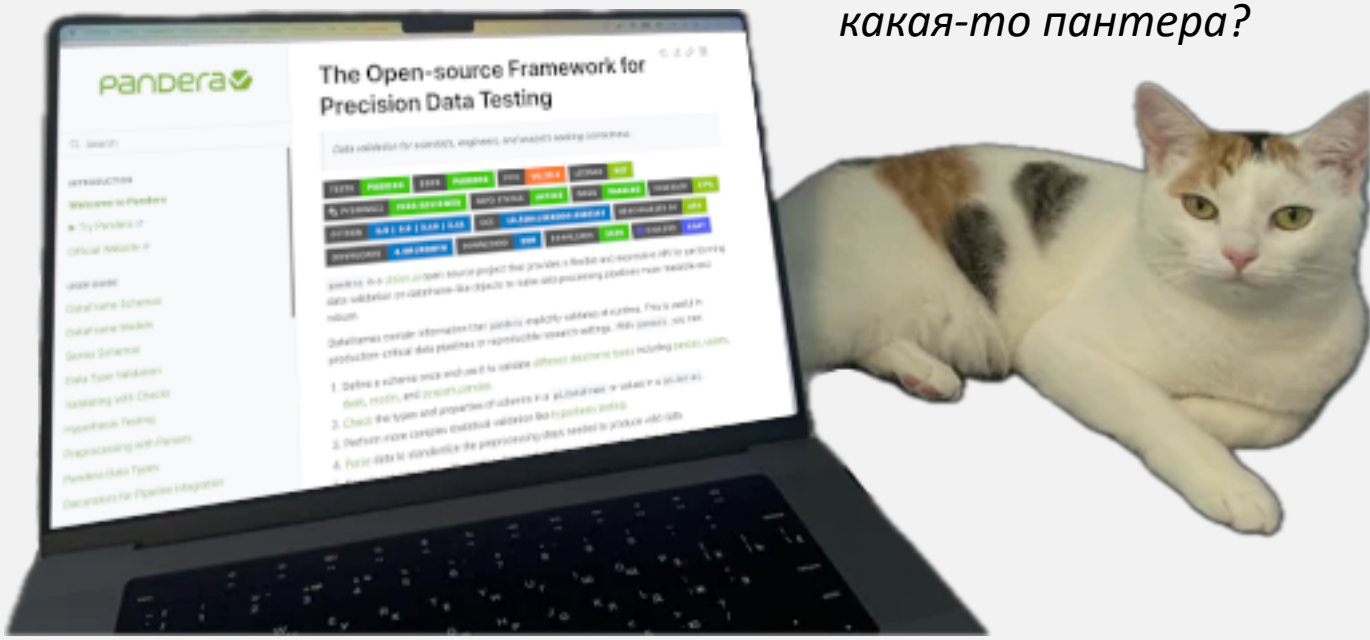
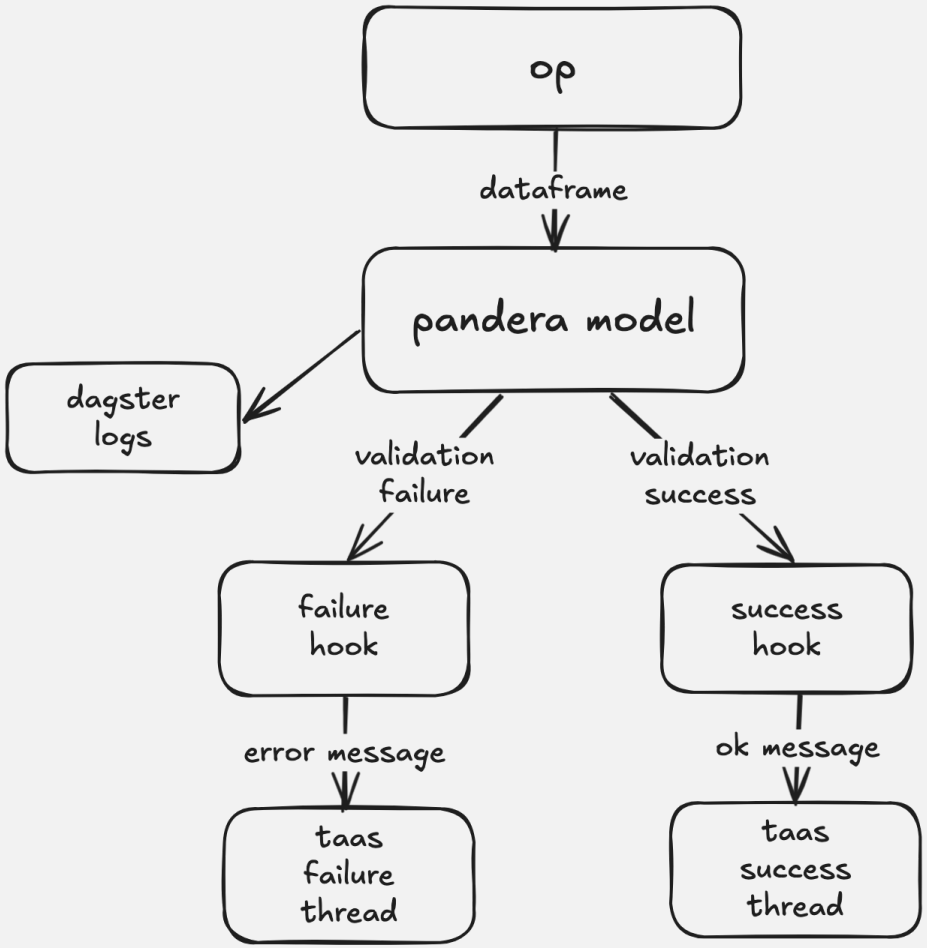


Схема с валидацией и уведомлением




```python
import pandera as pa
from pandera.typing import Series
from dagster_pandera import pandera_schema_to_dagster_type

class SomeDFWithGeo(pa.DataFrameModel):
    name: Series[str] = pa.Field(description='POI name')
    traffic: Series[int] = pa.Field(ge=0, nullable=False,
                                    description='Traffic nearby')
    geopos: Series[object] = pa.Field(nullable=False,
                                      description='Point geometry')

    @pa.check('geopos', name='Correct geojson')
    def check_geopos(col: Series[object]) -> bool:...

@op(out=Out(dagster_type=pandera_schema_to_dagster_type(SomeDFWithGeo)))
def process_spatial_data():...

@job
def spatial_job():
    process_spatial_data.with_hooks({taas_success_hook})()
```

Итоги внедрения

Для всех:

- своевременные уведомления о проблемах в данных

История про непривычно много данных

История про непривычно много данных

- Встретили задачу, которая считалась привычными инструментами в руках аналитика 5 месяцев
- Данные лежат в паркетах в S3

Откуда столько?!

Dagster + Spark

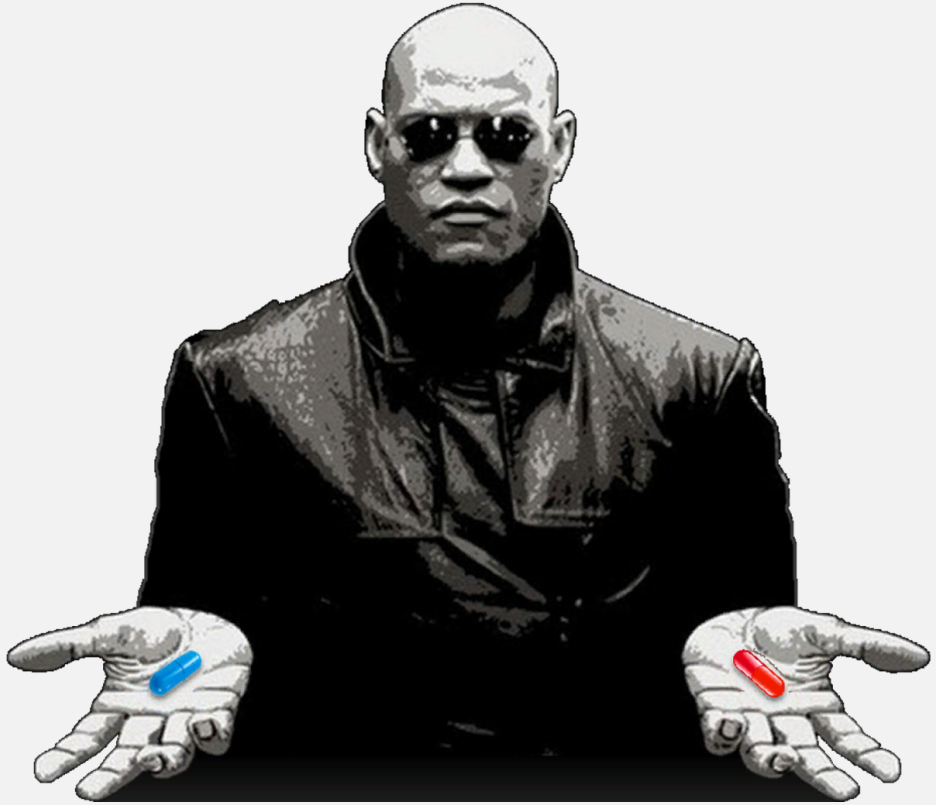
Интеграция с Apache Spark:

- Databricks
- AWS EMR
- YARN / spark standalone

Dagster + spark-operator

Проблемы, с которыми столкнулись при интеграции:

- сложности с запуском приложений через yaml конфиг
- необходимость отслеживать прогресс задачи
- хочется читать логи



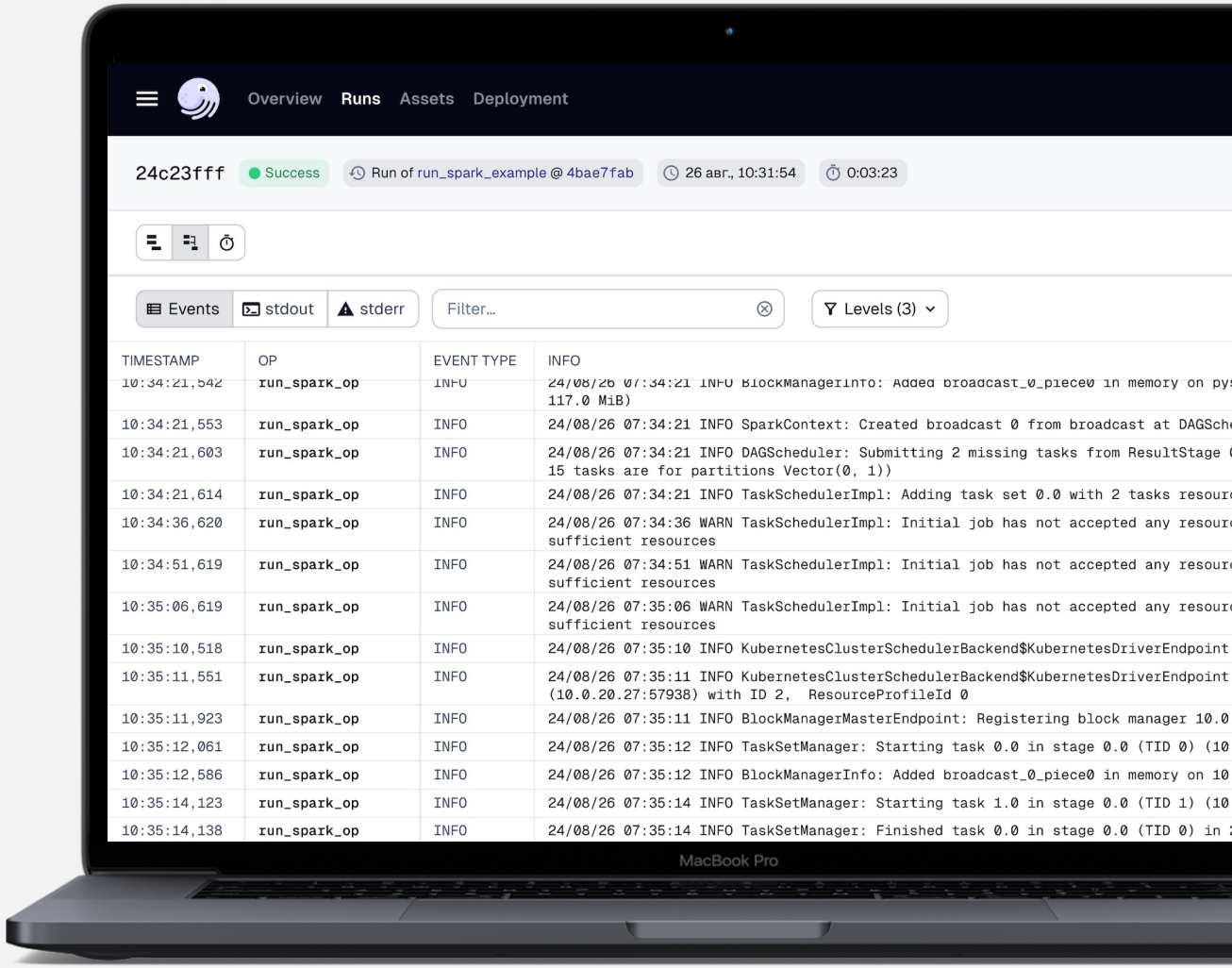
пишешь все сам

пишешь все сам, только клубничная

Dagster + spark-operator

Собственная библиотека:

- задание конфигурации с помощью кода
- запуск и очистка ресурсов в k8s
- стриминг логов драйвера



spark8s

https://github.com/bestplace/spark8s

```python
app_name = 'test-app'
# create manifest using static method
app = AppManifest.load_default_manifest(app_name)

app.set_driver_spec(1, '1000m', 2)
app.set_executor_spec(1, '1000m', 2, 4)
app.set_application_file('local:////somefile.py')

manifest = app.get_manifest()

# create kubernetes api client
api = client.ApiClient()
core_v1_api = client.CoreV1Api()
pod_name = f'{app_name}-driver'

# run app, stream logs, delete app and allocated resources
run_spark_app(api, manifest)
for line in stream_pod_logs(core_v1_api, pod_name, namespace):
    log.info(line)
delete_spark_app(api, app_name)
```

Итоги внедрения

Для аналитика:

- новый инструмент обработки данных
- в конкретной задаче кратное ускорение

Для дата-инженера:

- инфраструктура для запуска spark приложений
- библиотека для шаблонизации конфигов и управления запусками

История про аналитику и визуализации

История про аналитику и визуализации

Проблема:

- аналитика и графики в jupyter ноутбуке
- отсутствие версионирования
- неочевидная логика преобразований

Мне кажется, или это не best practice?

Как же автоматизировать пайплайны?

- Настройка выгрузки данных по расписанию
- Добавление автоматизации преобразований

Как же автоматизировать пайплайны?

- Настройка выгрузки данных по расписанию
- Добавление автоматизации преобразований

- Используем dbt и его интеграцию с dagster

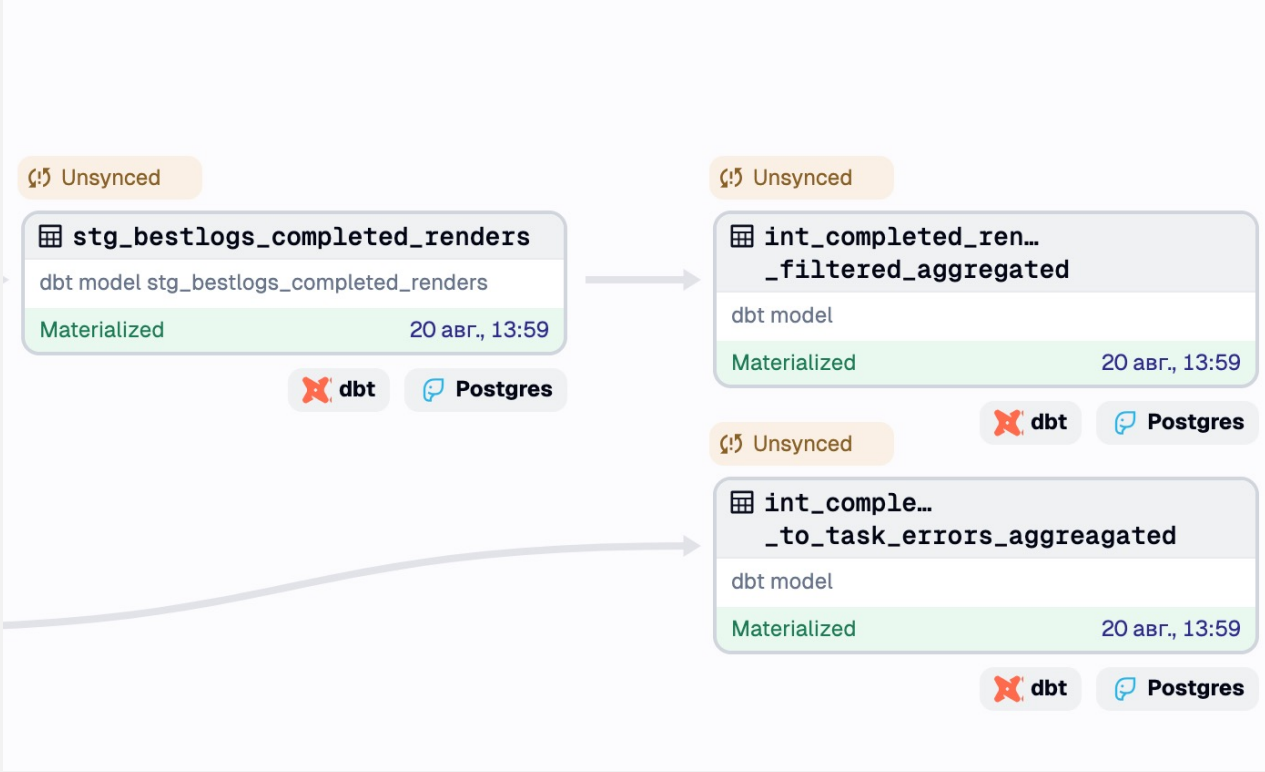
dagster + dbt

У дагстера есть **asset** — объект в постоянном хранилище (таблица, файл и т.п.)

У dbt есть **модели**

dagster-dbt дает нам:

- dbt_assets
- интеграцию в глобальный lineage
- возможность запуска и параметризация через UI



dagster + dbt

Проблемы, с которыми столкнулись при интеграции:

- ???

Итоги внедрения

Для менеджера:

- дашборды на свежих данных каждое утро
- не нужно заниматься поддержкой кода

Для дата-инженера:

- автоматизация создания витрин
- процессы работы с данными стали прозрачнее
- граф связей между данными из разных источников

Итоги

Итоги

- Легко вкатиться как дата-инженеру,
  так и аналитику
- Большое количество интеграций
- Активная поддержка со стороны разработчиков и растущее англоязычное коммьюнити
- Несложное развертывание и надежная работа

```
git clone https://github.com/dagster-io/dagster.git
cd dagster/examples/deploy_docker
docker compose up --build
```

Bestplace

Спасибо за внимание!

Дмитрий Крылов
mityakrylov

Алексей Завальский
al_zav

и Скрепка