

AIRFLOW 2.X SAAS



Солодягин Михаил

Юнк Сергей

Суханов Вадим

TELE2

ПЛАН ДОКЛАДА

1. Старая структура проекта и дагогенератор
2. Требования к новому проекту и причины выбора SaaS решения
3. Архитектура нового проекта
4. Реализация Secrets backend
5. Gitlab CI и поветочный деплой
6. Мониторинг и логи, время жизни стенда, хранилище
7. Подводные камни и проблемы
8. Демонстрация
9. Результаты проекта Airflow SaaS
10. Q&A

AIRFLOW – ЭТО...

01.

Open source инструмент для создания, оркестрации и мониторинга процессов написанный на Python

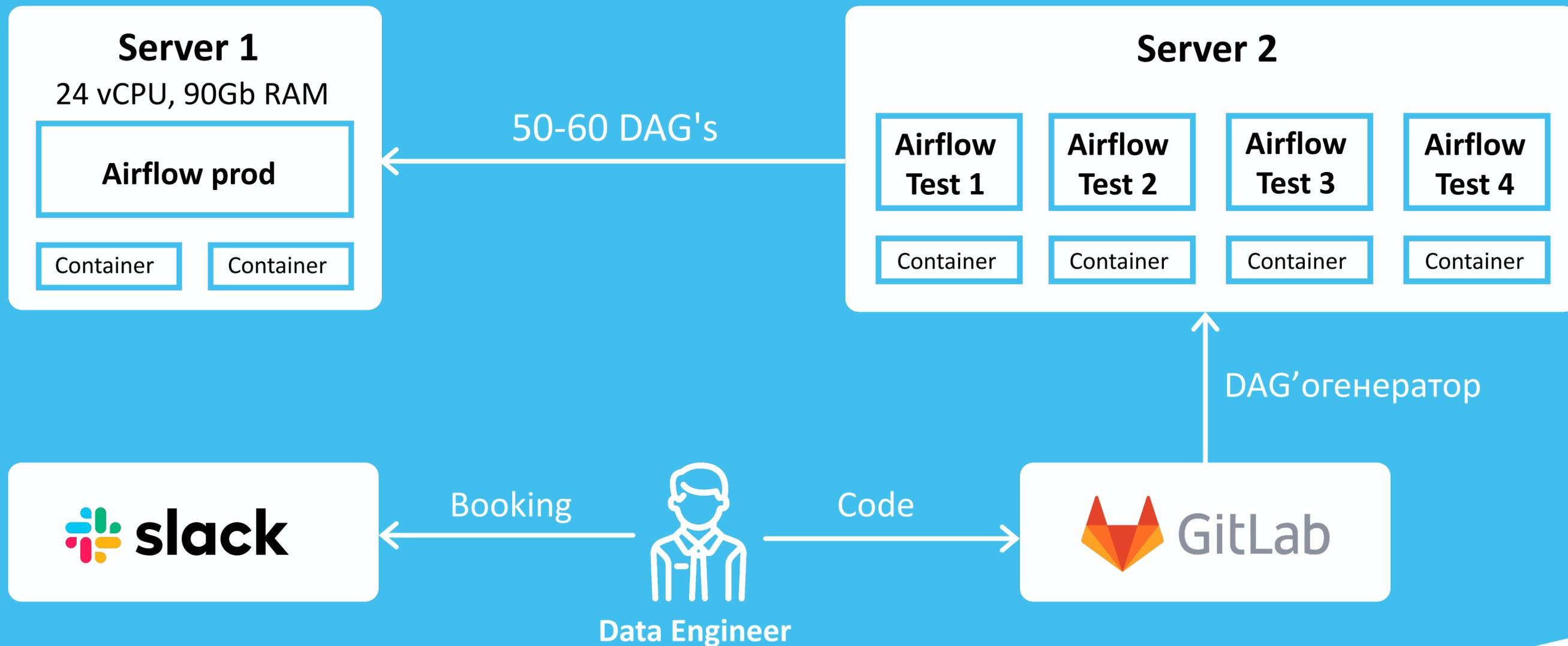
02.

Процессы описываются в виде операторов которые объединяются в направленные ациклические графы, сокращенно DAG.

03.

Используем для запуска регулярных процессов по перегрузке данных и запуска регулярных расчетов

СТАРАЯ СТРУКТУРА ПРОЕКТА И DAG'ОГЕНЕРАТОР



ПРИМЕР КОНФИГУРАЦИИ ДЛЯ DAG'ОГЕНЕРАТОРА

```
dags:
  dwh_daily_dicts:
    owner: DE
    schedule_interval: 01 18 * * * } Интервал и время запуска
    start_date: '20200902'
    tasks:
      load_table:
        type: k8s_jdbc_to_hdfs } Тип задачи
        size: XS
        jdbc:
          connection: connection_name } База данных А
          db: db_name
          table: table_name
        hdfs:
          location: "/path/to/hdfs" } Путь до Hadoop В
      abc_regions_loader: } Расчет данных С
        type: k8s_spark
        driver:
          ...
        executor:
          ...
```

1

```
...
  deploy_mode: cluster
  master: yarn
  java_class: 'com.tele2.tdp.abc.AbcRegionsLoader'
  jars: /extra/*.jar,/etl.jar
  name: abc_regions_loader
  application: /etl.jar
  typesafe_config:
    source:
      rawAbcPath: "/path/to/hdfs/{{ds}}"
      regionMincomPath: "/regions_mincom/{{ds}}"
    sink:
      path: "/abc_regions"
  image: tele2-artifactory/abc-regions-loader:5-14
  flow:
    - load_table
    - abc_regions_loader } Очередность исполнения
```

2

Конфигурация
Spark-задачи

«БУКИНГ» СВОБОДНОГО СТЕНДА

test-environment-booking ▾

+ Add a bookmark

Wednesday, June 30th ▾

 Vladimir Takhmazyan 3:14 PM
0 отпустил

 Vladimir Takhmazyan 3:42 PM
Занял 0

 Vladimir Takhmazyan 9:39 PM
Отпустил

 Andrey Kupin 9:39 PM
Займу 0

 Andrey Kupin 11:02 PM
Отпустил

Thursday, July 1st ▾

 Andrey Kupin 3:04 PM
Займу 0

 Andrey Kupin 4:24 PM
Отпустил

 Vladimir Takhmazyan 6:53 PM
Занял 0

Friday, July 2nd ▾

 Vladimir Takhmazyan 4:52 PM
Отпустил

 Vladimir Takhmazyan 10:44 PM
Никто не против, если на все выходные тестовый займу?

Tuesday, July 6th ▾

 Andrey Kupin 6:29 PM
0 airflow занят?

 Vladimir Takhmazyan 6:29 PM
Мной пока нет

 Andrey Kupin 6:30 PM
Когда нужен будет?

 Vladimir Takhmazyan 6:31 PM
Как закончишь 😊

 Andrey Kupin 6:32 PM
Ок, займу)

 Andrey Kupin 7:10 PM
Отпустил

ИСТОРИЯ ОДНОЙ РАЗРАБОТКИ...

Случайная поломка
тестового стенда или долгое
исполнение задач на нем
влияет на остальных
разработчиков



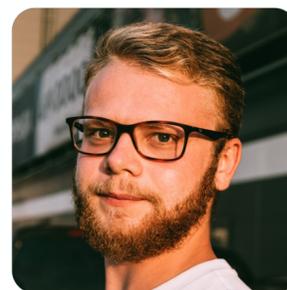
Руководитель 12:00

Иван, привет! Подскажи, что там с новым DAG'ом, у нас по плану сегодня релиз?



Разработчик 12:02

Мне нужен именно тот стенд, на котором я вбивал свои пароли!



Разработчик 12:03

А он, как назло, сломался!

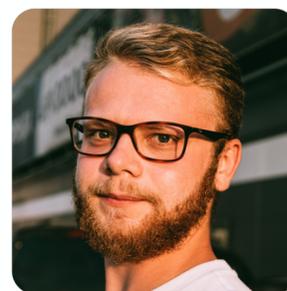
ИСТОРИЯ ОДНОЙ РАЗРАБОТКИ...

Логи хранятся на той же
машине, где и развернут
тестовый стенд.
К ней нет доступа у рядового
разработчика



Руководитель 12:00

Иван, привет! Подскажи, что там с новым DAG'ом, у нас по плану сегодня релиз?



Разработчик 12:02

Мне нужен именно тот стенд, на котором я вбивал свои пароли!

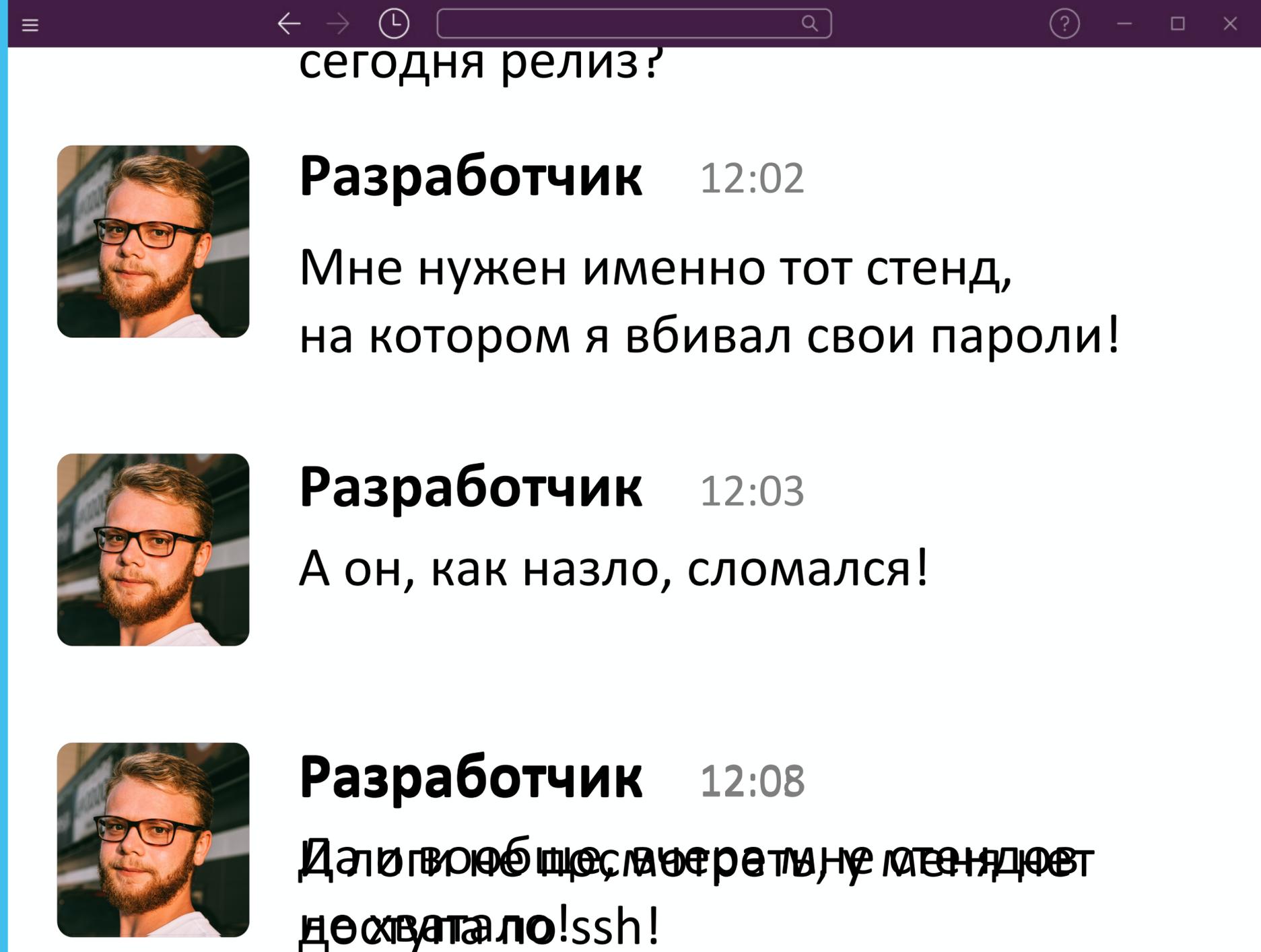


Разработчик 12:03

И логи не посмотреть, у меня нет доступа по ssh!

ИСТОРИЯ ОДНОЙ РАЗРАБОТКИ...

Создание нового стенда – это история на несколько часов, пока освободится администратор и вручную поднимет еще один экземпляр тестового стенда



сегодня релиз?

 **Разработчик** 12:02
Мне нужен именно тот стенд, на котором я вбивал свои пароли!

 **Разработчик** 12:03
А он, как назло, сломался!

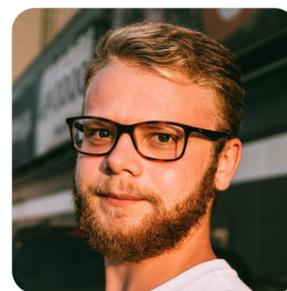
 **Разработчик** 12:08
Дали вообще серверу не стэндост
де хвотало!ssh!

ИСТОРИЯ ОДНОЙ РАЗРАБОТКИ...

При этом в случае, когда
стенд уже не нужен, он все
еще продолжит существовать
на машине, пока кто-нибудь
не соберется его удалить



Разработчик 12:03
А он, как назло, сломался!



Разработчик 12:05
И логи не посмотреть, у меня нет
доступа по ssh!



Разработчик 12:08
Дага два вообще, вчера же не служило
не хватало!

ИСТОРИЯ ОДНОЙ РАЗРАБОТКИ...

Из-за этих проблем сильно
страдает ТТМ по задачам



Разработчик 12:05

И логи не посмотреть, у меня нет доступа по ssh!



Разработчик 12:08

Да и вообще, вчера мне стендов не хватало!



Разработчик 12:09

~~В будущем мне за это же~~ не нужны!
я не смог вовремя вывести
новый бизнес-функционал

МЫ РЕШИЛИ

**ХВАТИТ
ЭТО
ТЕРПЕТЬ!**



ТРЕБОВАНИЯ К НОВОМУ ПРОЕКТУ

Решает все старые
проблемы

Умеет во временные
хранилища

Поддерживает кастомные
образы с разными наборами
библиотек

Возможность быстрой
смены Executor's

Self-Hosted

ПРИЧИНЫ ВЫБОРА SAAS РЕШЕНИЯ

Альтернативы

Создать стенд под каждого из разработчиков

- Изолированность, свой набор секретов

Ansible Playbook

- Для подготовки стенда больше не нужны админы
- Изолированные стенды

Переход на другое решение

- Возможно, мы попадаем в страну единорогов с радугой

Плюсы

Минусы

- Не решает архитектурных проблем
- Отсутствует возможность работать сразу с двумя и более фичами
- Секреты все еще нуждаются в синхронизации

- Преаллокация ресурсов
- Нет доступа к логам
- Сложность с доставкой секретов
- Нет временного хранилища
- Отсутствие возможности быстро сменить Executor

- Возможно попадем...
А может, и нет...
- Ah shit, here we go again...

ПРЕИМУЩЕСТВА SAAS

Решает
все прошлые
проблемы

Удовлетворяет
всем заявленным
требованиям

Универсальное
решение,
подходящее для
всех команд

АРХИТЕКТУРА НОВОГО ПРОЕКТА



TELE2

ТРЕБОВАНИЯ К КОМПОНЕНТАМ

- | Преимущественно opensource решения
- | Нативная поддержка Airflow
- | Простота обслуживания

ВЫБРАННЫЕ КОМПОНЕНТЫ



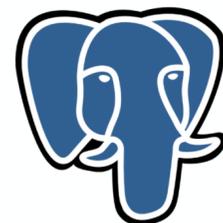
kubernetes

Платформа



elasticsearch

Логи



PostgreSQL

База данных



ceph

Хранилище



LDAP

Авторизация



Prometheus



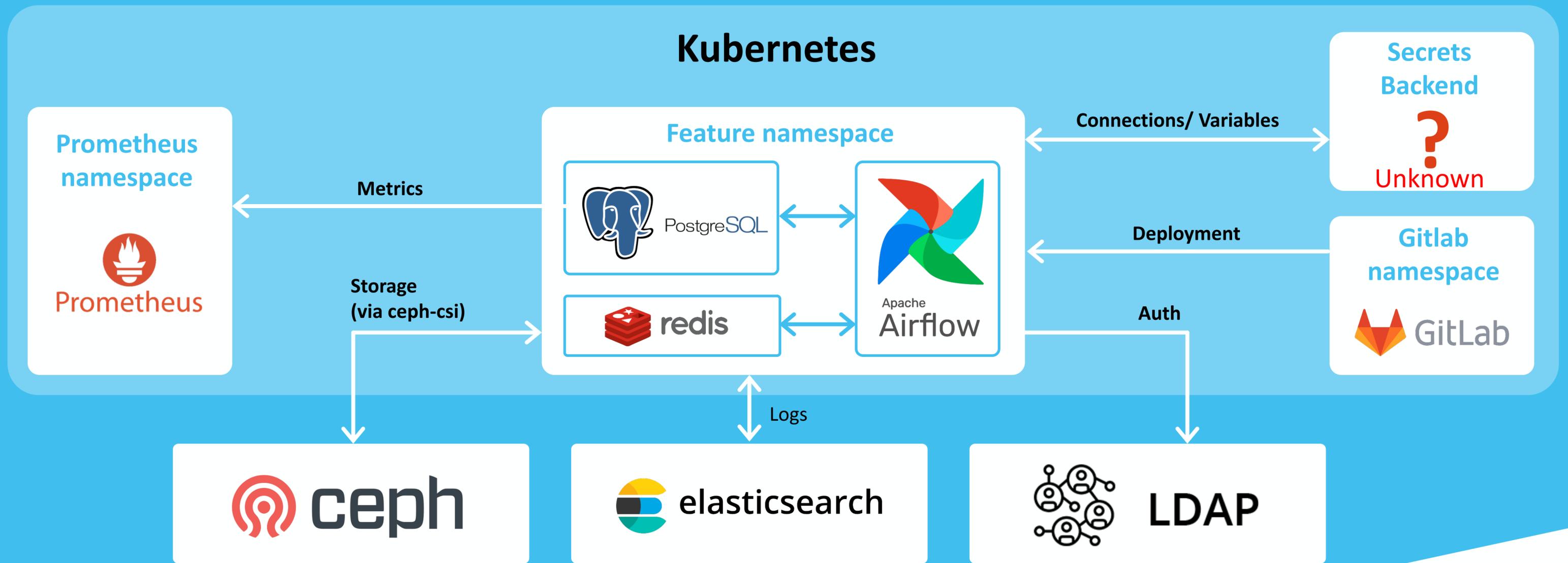
STATSD



Grafana

Мониторинг

СХЕМА НОВОЙ АРХИТЕКТУРЫ



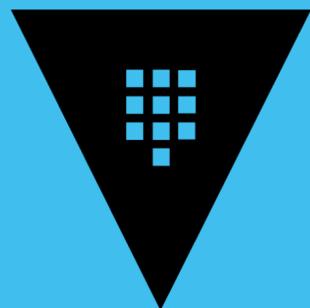
ТРЕБОВАНИЯ К ХРАНЕНИЮ СЕКРЕТОВ

| **Изолированность**

| **Наследуемость**

| **Безопасность**





HashiCorp

Vault

РЕАЛИЗАЦИЯ
SECRETS BACKEND



TELE2

КАК VAULT PROVIDER ЗАБИРАЕТ СЕКРЕТЫ

```
def get_conn_uri(self, conn_id):  
    ...  
    response = self._get_secret(self.connections_path, conn_id)  
    return response.get("conn_uri") if response else None  
} Connections  
  
def get_variable(self, key):  
    ...  
    response = self._get_secret(self.variables_path, key)  
    return response.get("value") if response else None  
} Variables  
  
def _get_secret(self, path_prefix, secret_id):  
    ...  
    secret_path = self.build_path(path_prefix, secret_id) } Формируем путь до секрета  
  
    try:  
        ...  
        response = self.client.secrets.kv.v2.read_secret_version(  
            path=secret_path, mount_point=self.mount_point) } Забираем секреты  
    except InvalidPath:  
        self.log.info("Secret %s not found in Path: %s", secret_id, secret_path)  
        return None  
    ...
```

ИЕРАРХИЯ СЕКРЕТОВ

airflow/master/<project_namespace_*>

Hashicorp Vault

Project Namespace A

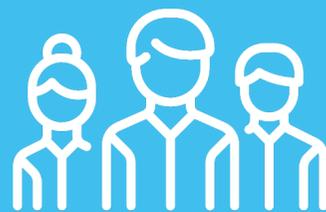
Connections, variables,
keytab, certificates

Read
Write



Lead A

Read
Only



Team A

Project Namespace B

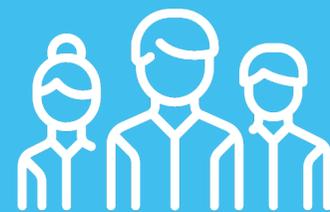
Connections, variables,
keytab, certificates

Read
Write



Lead B

Read
Only



Team B

Project Namespace C

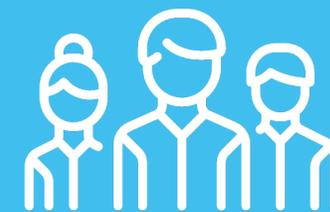
Connections, variables,
keytab, certificates

Read
Write



Lead C

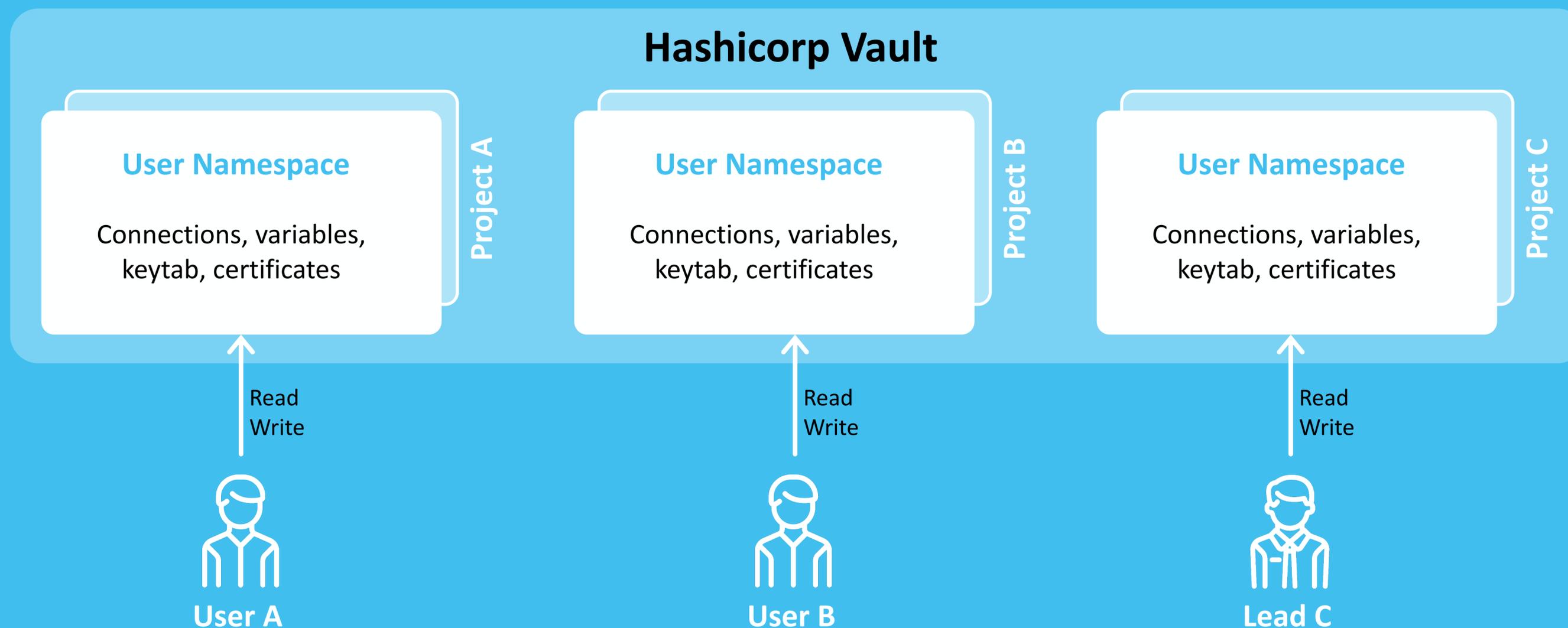
Read
Only



Team C

ИЕРАРХИЯ СЕКРЕТОВ

airflow/<user_*>/<project_namespace_*>



РЕАЛИЗАЦИЯ В КОДЕ ПРОВАЙДЕРА

```
def build_path_tele2(self, scope: str, path: str, key: str): } Создаем путь до секрета
    return f"{scope}/{path}/{key}"

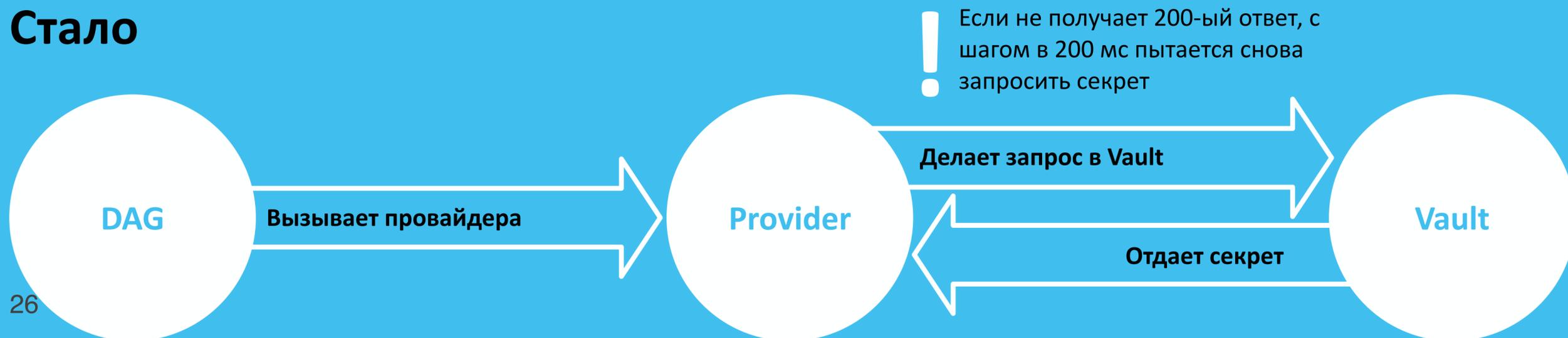
def _get_secret(self, path_prefix, secret_id):
    ...
    try:
        secret_path = self.build_path_tele2(
            os.environ["GITLAB_USER_LOGIN"], } Забираем имя пользователя из переменной окружения
            path_prefix,
            secret_id)
        response = self.client.secrets.kv.v2.read_secret_version( } Пытаемся получить секрет
            path=secret_path, mount_point=self.mount_point)                из userspace
    except:
        try:
            secret_path = self.build_path_tele2(self.scope, path_prefix, secret_id) } Пытаемся получить
            response = self.client.secrets.kv.v2.read_secret_version(                секрет из
                path=secret_path, mount_point=self.mount_point)                projectspace
        except InvalidPath:
            self.log.info("Secret %s not found in Path: %s", secret_id, secret_path)
            return None
    ...
```

ПРОБЛЕМЫ С SECRET'S BACKEND

Было



Стало



ПРОБЛЕМЫ С SECRET'S BACKEND

```
def get_connection(self, conn_id: str) -> Optional['Connection']:  
    """  
    Return connection object with a given ``conn_id``.  
    :param conn_id: connection id  
    :type conn_id: str  
    """  
    from airflow.models.connection import Connection  
  
    conn_uri = self.get_conn_uri(conn_id=conn_id) } Получает conn_uri в виде строки:  
    if not conn_uri: } my-conn-type://login:password@host:port/schema?params  
        return None  
    conn = Connection(conn_id=conn_id, uri=conn_uri) } Создает объект Connection  
    return conn
```

ПРОБЛЕМЫ С SECRET'S BACKEND

```
class Connection(Base, LoggingMixin):
```

Класс

```
...
def __init__(
    self,
    ...
):
    super().__init__()
    self.conn_id = conn_id
...
if uri:
    self._parse_from_uri(uri)
else:
    self.conn_type = conn_type
    self.host = host
    self.login = login
    self.password = password
    self.schema = schema
    self.port = port
    self.extra = extra
```

Если передан conn_uri
пытаемся его парсить

```
def _parse_from_uri(self, uri: str):
```

Парсер

```
uri_parts = urlparse(uri)
conn_type = uri_parts.scheme
if conn_type == 'postgresql':
    conn_type = 'postgres'
elif '-' in conn_type:
    conn_type = conn_type.replace('-', '_')
self.conn_type = conn_type
self.host = _parse_netloc_to_hostname(uri_parts)
quoted_schema = uri_parts.path[1:]
self.schema = unquote(quoted_schema) if quoted_schema else quoted_schema
self.login = unquote(uri_parts.username) \
    if uri_parts.username else uri_parts.username
self.password = unquote(uri_parts.password) \
    if uri_parts.password else uri_parts.password
self.port = uri_parts.port
if uri_parts.query:
    self.extra = json.dumps(dict(parse_qs(uri_parts.query, keep_blank_values=True)))
```

При попытке распарсить строку
jdbc:teradata://10.12.76.212/charset=UTF8
отдаст пустые поля

ПРОБЛЕМЫ С SECRET'S BACKEND

```
def get_connection(self, conn_id: str) -> Optional['Connection']:
    """
    Return connection object with a given ``conn_id``.

    :param conn_id: connection id
    :type conn_id: str
    """
    from airflow.models.connection import Connection

    conn_uri = self.get_conn_uri(conn_id=conn_id)
    if not conn_uri:
        return None
    #conn = Connection(conn_id=conn_id, uri=conn_uri) - Будем забирать сразу Json из Vault
    conn = Connection(conn_id=conn_id, **conn_uri)
    return conn
```

Решение

ЧТО В ИТОГЕ ПОЛУЧИЛОСЬ?

[airflow](#) < [mikhail.solodyagin](#) < [TDP-AIRFLOW](#) < [connections](#) < [teradata_dev](#)

Create new version

JSON

Secret metadata

Maximum Number of Versions

10

Require Check and Set ⓘ

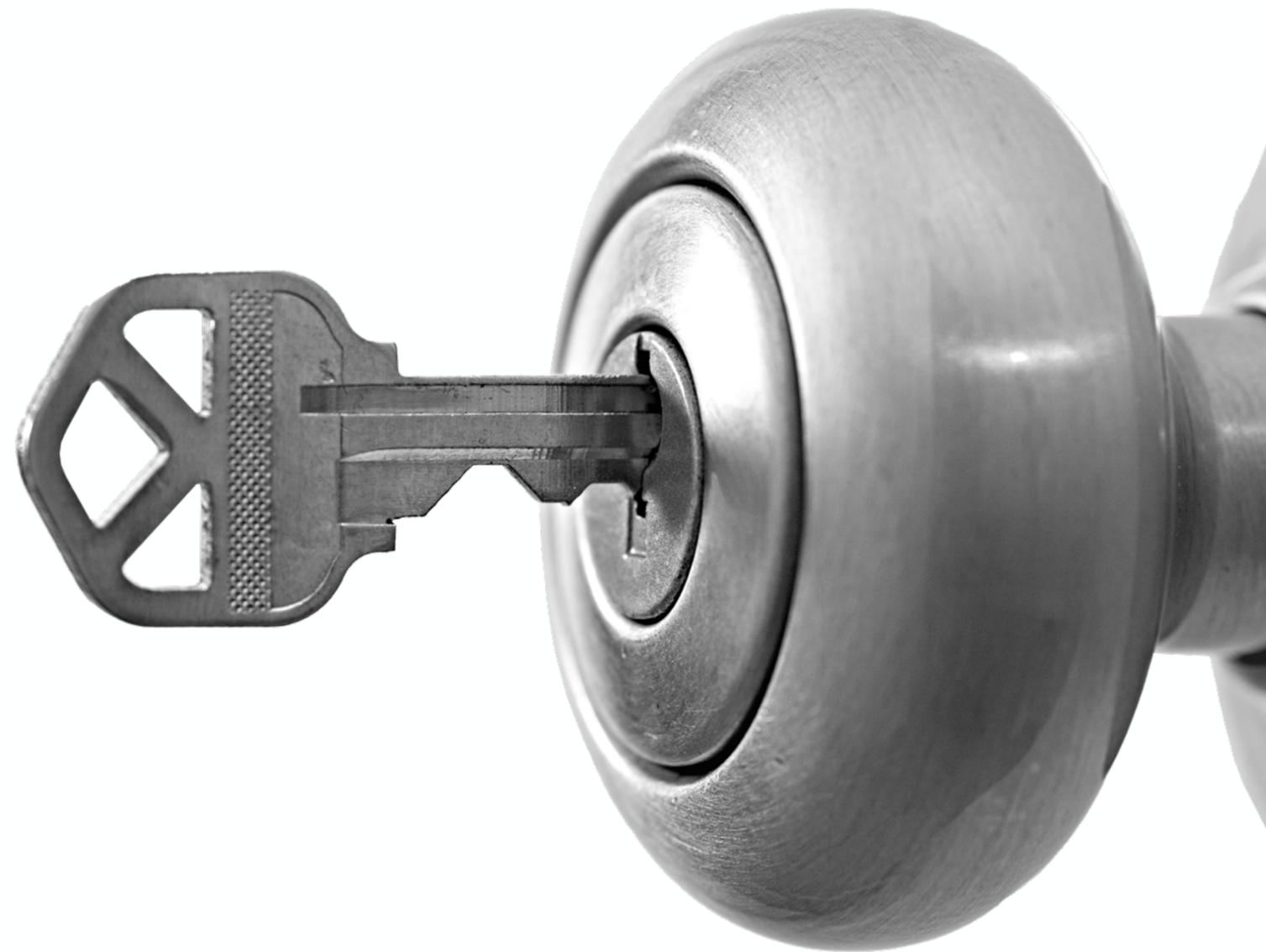
Version data

conn_type	jdbc		
extra	{"extra_jdbc_drv_path": "/opt/airflow/jars/terajdbc4.jar", "extra_jdbc_drv_clsname": "com.teradata.jdbc.TeraDriver"}		
host	jdbc:teradata://10.12.76.212/charset=UTF8		
login	mikhail.solodyagin		
password	supersecret!@#123		
key	value		Add

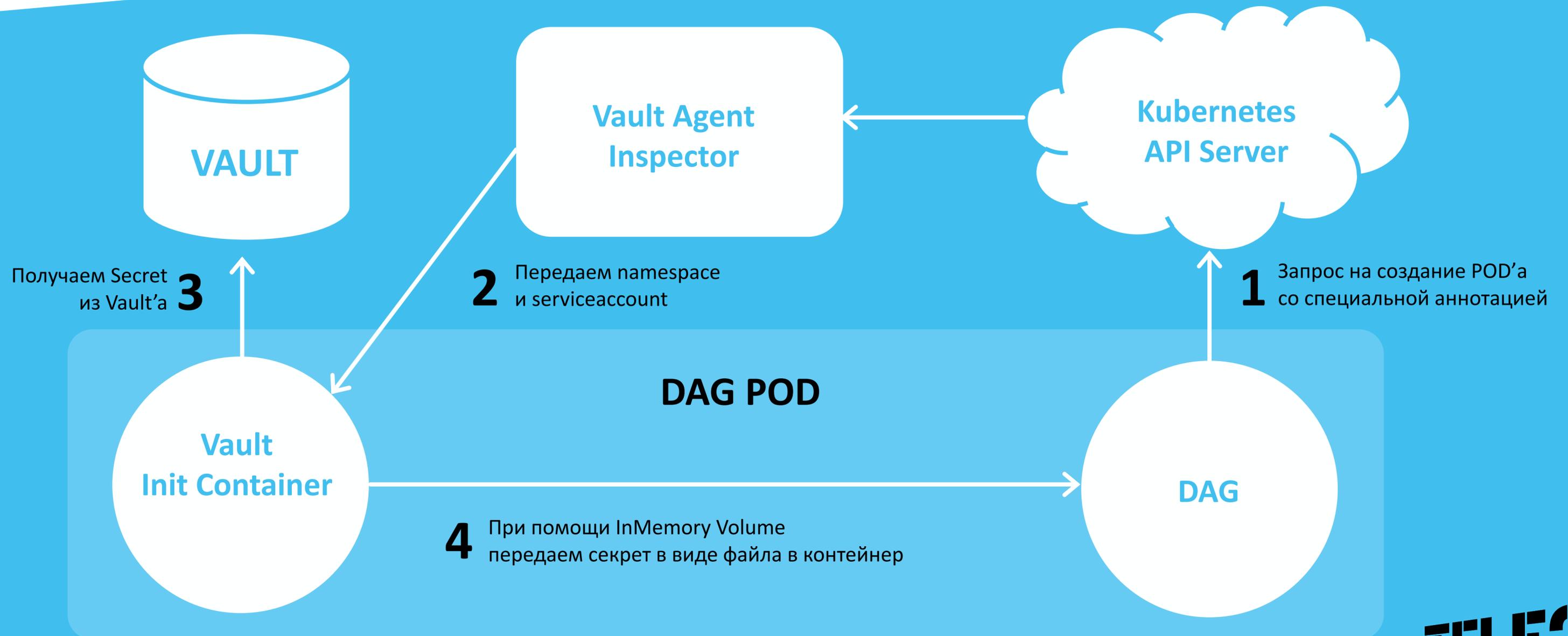
Save

Cancel

Ssh-ключи, keytab'ы и секретные файлы

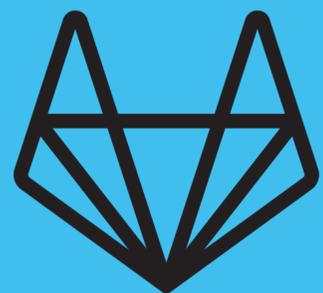


РЕШЕНИЕ: VAULT AGENT INJECTOR



ПРИМЕР АННОТАЦИИ, ДОБАВЛЯЕМОЙ В POD

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    vault.hashicorp.com/agent-inject: "true"
    vault.hashicorp.com/role: "tdp_airflow_prod_read_role" } Роль на сервере Hashicorp Vault
    vault.hashicorp.com/agent-pre-populate-only: "true"
    vault.hashicorp.com/agent-inject-file-keytab-int_01: int_01.keytab } Имя файла и путь
    vault.hashicorp.com/secret-volume-path-keytab-int_01: /opt/keytabs/ } до него внутри пода
    vault.hashicorp.com/agent-inject-template-keytab-int_01: | } Шаблон, описывающий
      {{- with secret "airflow/master/tdp-airflow-prod/keytabs" -}} } сохранение секрета
      {{- base64Decode .Data.data.int_01 -}} } в файл
      {{- end -}}
    vault.hashicorp.com/agent-inject-secret-keytab-int_01: | } Путь до секрета
      airflow/master/tdp-airflow-prod/keytabs
```



GitLab

**GITLAB CI
И ПОВЕТОЧНЫЙ
ДЕПЛОЙ**



TELE2

СХЕМА ПРИВЯЗКИ СТЕНДА К ВЕТКАМ

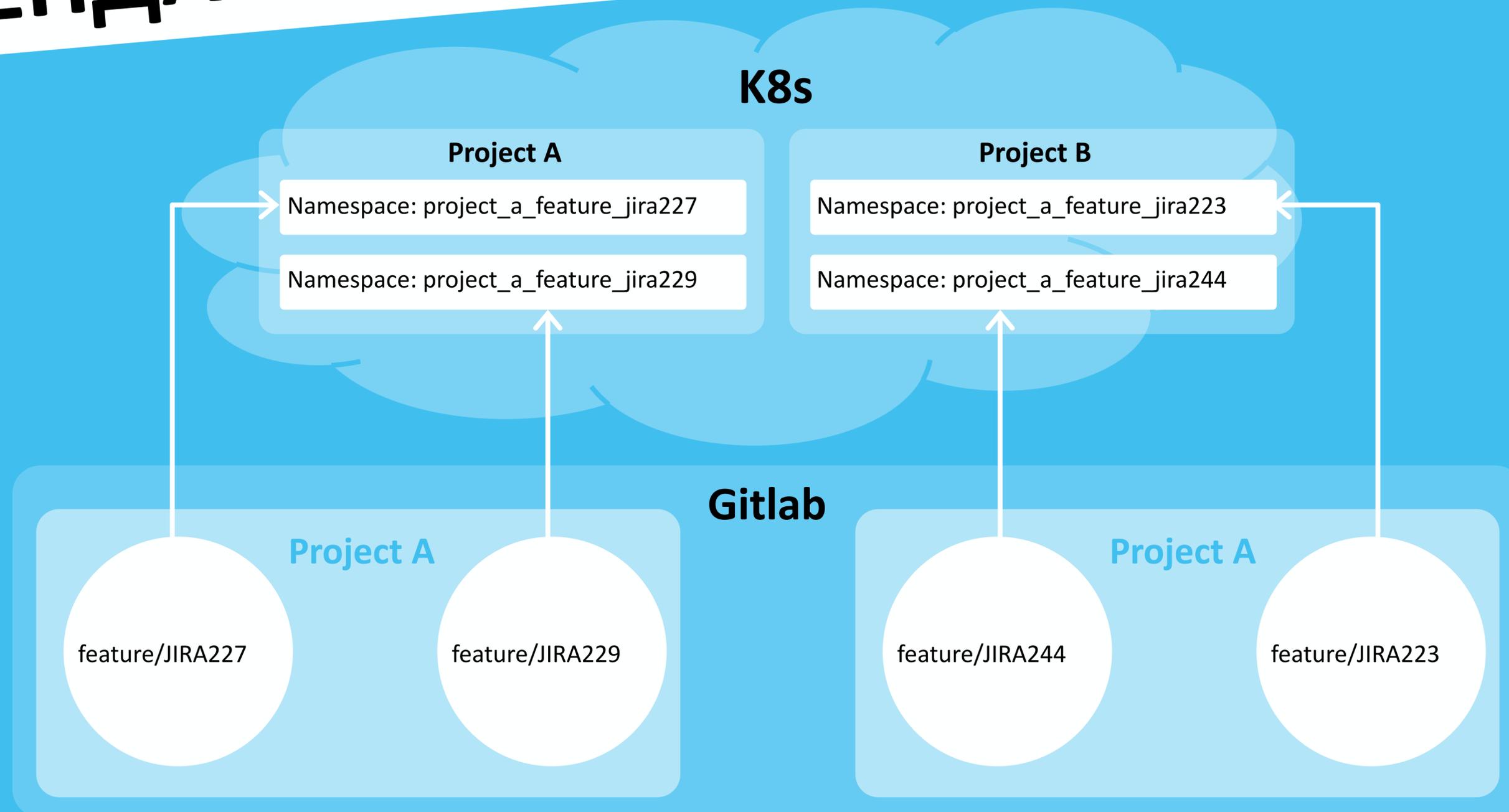


СХЕМА ПОСТАВКИ

Gitlab CI

- Подготавливаем дополнительные values и манифесты для k8s
- Определяем входные условия для деплоя
- Применяем манифесты к k8s (создаем namespace, configmap'ы, secret'ы)



Helm

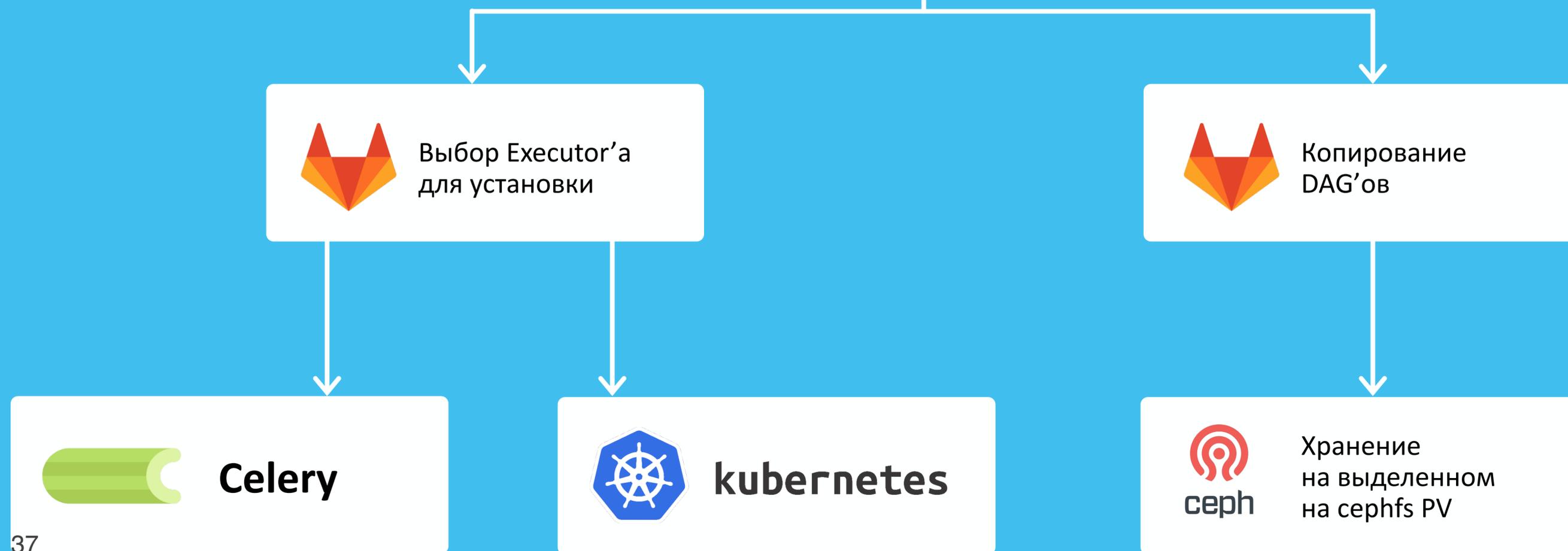
- Производит установку Airflow при помощи нашего helm-чарта
- Если Airflow уже установлен – производит его Upgrade, если Values/



k8s

- Поднимает внутренние объекты согласно примененным к нему манифестам
- Творит свою особую, «штурвальную» магию

ИТОГОВЫЙ PIPELINE



ДОСТАВКА DAG'ОВ РАНЬШЕ

Артефакт

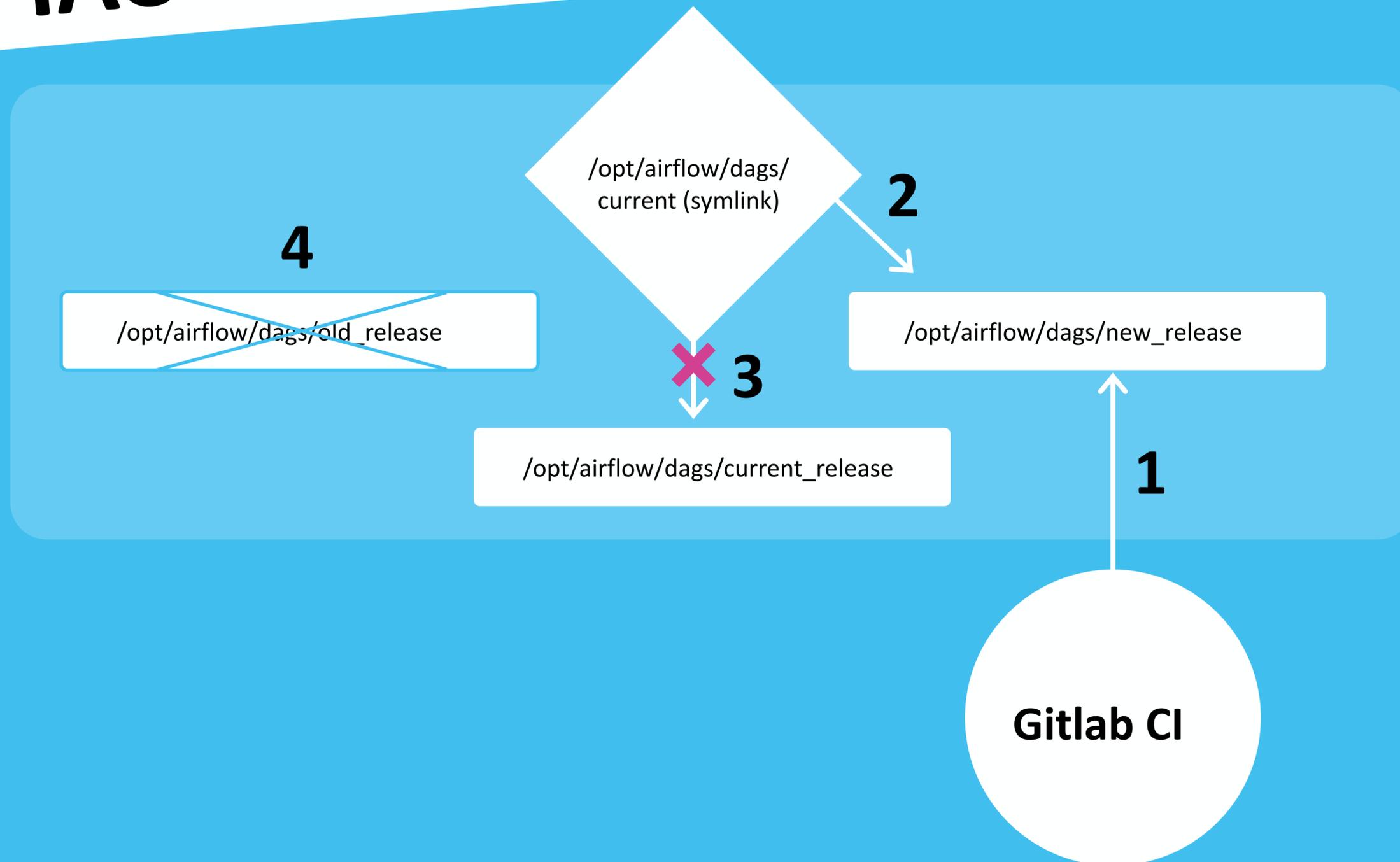
Zip-архив `airflow_dags.zip`, который собирается по коммиту после прохождения всех тестов



Копирование

При помощи `kubectl` сс копируем архив на `persistent volume` через `airflow-worker pod` в папку `/opt/airflow/dags/`, перезаписывая предыдущий

ДОСТАВКА DAG'ОВ СЕЙЧАС



А ПОЧЕМУ НЕ GIT-SYNC SIDECAR?

01.

Контролируемая
поставка вместо
«Случайный коммит»

02.

Нет необходимости менять
схему хранения в репе

03.

Простота
обслуживания для
Ops'ов

МОНИТОРИНГ,
ВРЕМЯ ЖИЗНИ
СТЕНДА,
ХРАНИЛИЩЕ



TELE2

ПРОБЛЕМЫ С ЛОГГИРОВАНИЕМ

```
[2021-10-11 11:26:28,164] {k8s_operator.py:138} INFO - Start create pvc: dagrun-shared-pvc-c575f39e58bdce5c1aeb1bc202dc109f
[2021-10-11 11:26:28,204] {k8s_operator.py:160} INFO - Created pvc: dagrun-shared-pvc-c575f39e58bdce5c1aeb1bc202dc109f
[2021-10-11 11:26:29,324] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:30,340] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:33,393] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:35,427] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:35,428] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:37,470] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:40,534] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:44,652] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Succeeded
[2021-10-11 11:26:44,666] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Succeeded
[2021-10-11 11:26:27,797] {taskinstance.py:876} INFO - Dependencies all met for <TaskInstance: pod_smartdata_demo_0_12_A_A_A.k8spod_echo_ls_smartdata_before_1_1 2021-10-09T12:00:00+00:00 [queued]>
[2021-10-11 11:26:27,810] {taskinstance.py:1087} INFO - Executing <Task(K8sPodOperator): k8spod_echo_ls_smartdata_before_1_1> on 2021-10-09T12:00:00+00:00
[2021-10-11 11:26:27,824] {standard_task_runner.py:77} INFO - Job 4: Subtask k8spod_echo_ls_smartdata_before_1_1
[2021-10-11 11:26:27,958] {logging_mixin.py:104} INFO - Running <TaskInstance: pod_smartdata_demo_0_12_A_A_A.k8spod_echo_ls_smartdata_before_1_1 2021-10-09T12:00:00+00:00 [running]> on host ***-worker-0.***-work
[2021-10-11 11:26:28,219] {k8s_operator.py:189} INFO - creating pod with labels {'dag_id': 'pod_smartdata_demo_0_12_A_A_A', 'task_id': 'k8spod_echo_ls_smartdata_before_1_1', 'execution_date': '2021-10-09T12000000
[2021-10-11 11:26:30,339] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:32,374] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:33,394] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:34,413] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:34,413] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:36,444] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:37,469] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Pending
[2021-10-11 11:26:38,497] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:40,535] {pod_launcher.py:126} WARNING - Pod not yet started: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7
[2021-10-11 11:26:41,558] {pod_launcher.py:189} INFO - Event: 70a61775456ae0689180352539158129.d702a782b5eb44d5b60dcb94b9caa3d7 had an event of type Running
[2021-10-11 11:26:27,771] {taskinstance.py:876} INFO - Dependencies all met for <TaskInstance: pod_smartdata_demo_0_12_A_A_A.k8spod_echo_ls_smartdata_before_1_1 2021-10-09T12:00:00+00:00 [queued]>
[2021-10-11 11:26:27,798] {taskinstance.py:1068} INFO - Starting attempt 1 of 2
[2021-10-11 11:26:27,798] {taskinstance.py:1069} INFO -
```

РЕШЕНИЕ

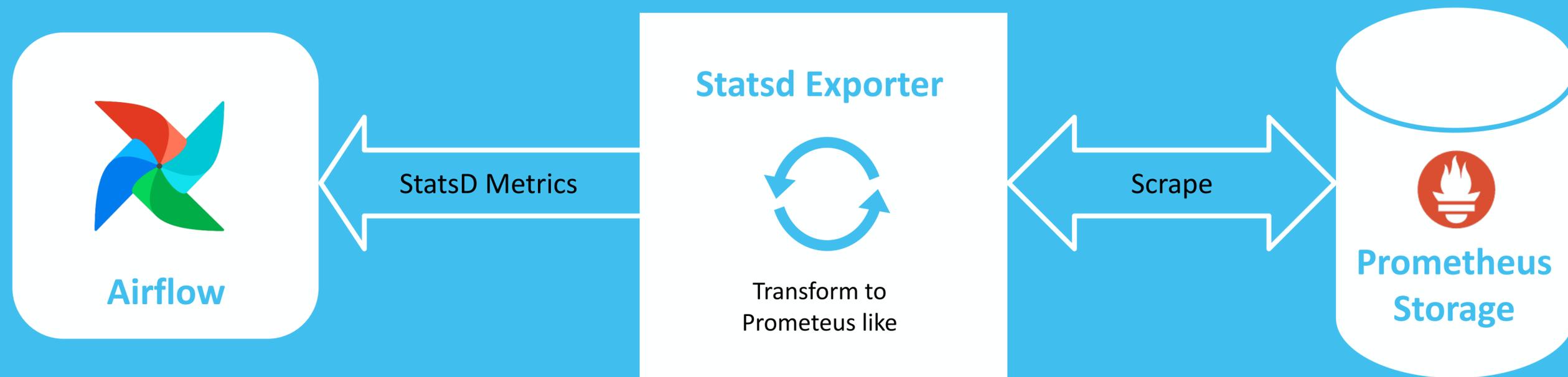
Фикс, ожидаем в будущих версиях Airflow

<https://github.com/apache/airflow/issues/17512>

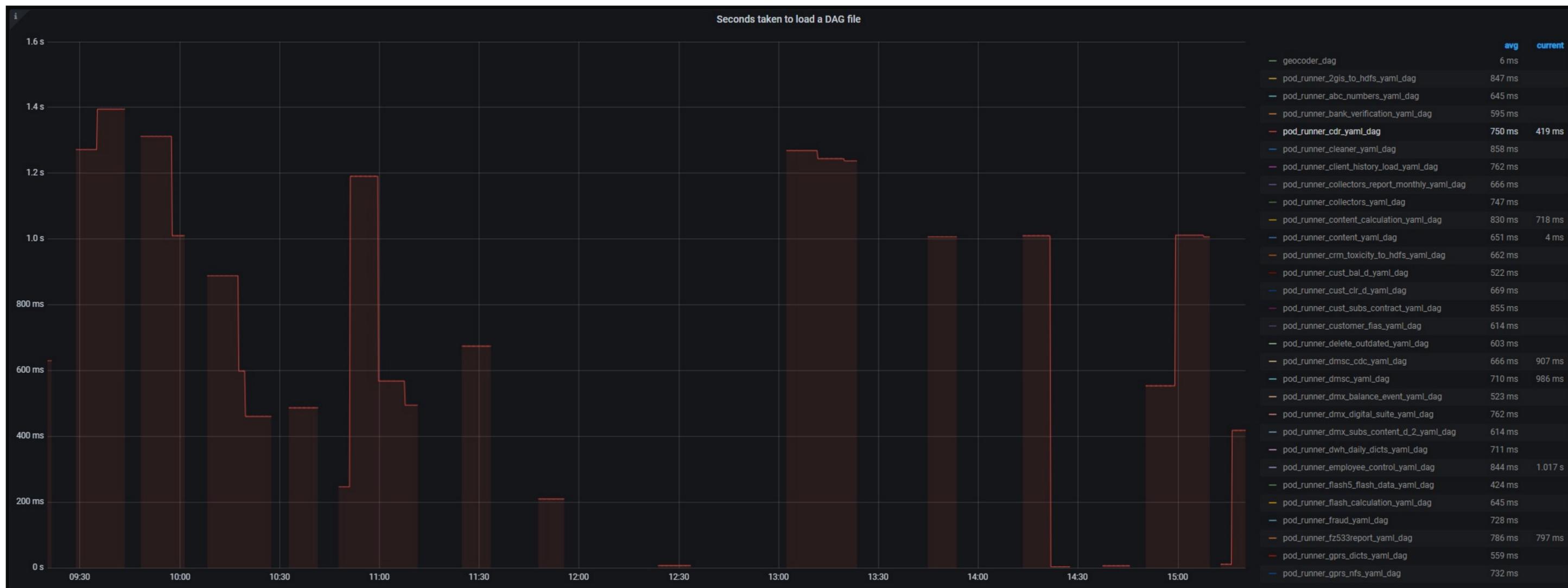
<https://github.com/apache/airflow/pull/17551>



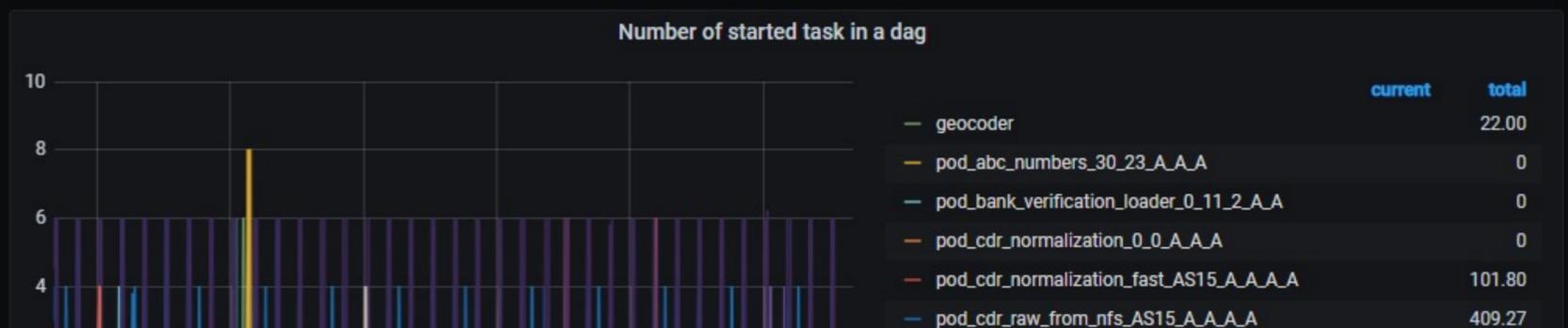
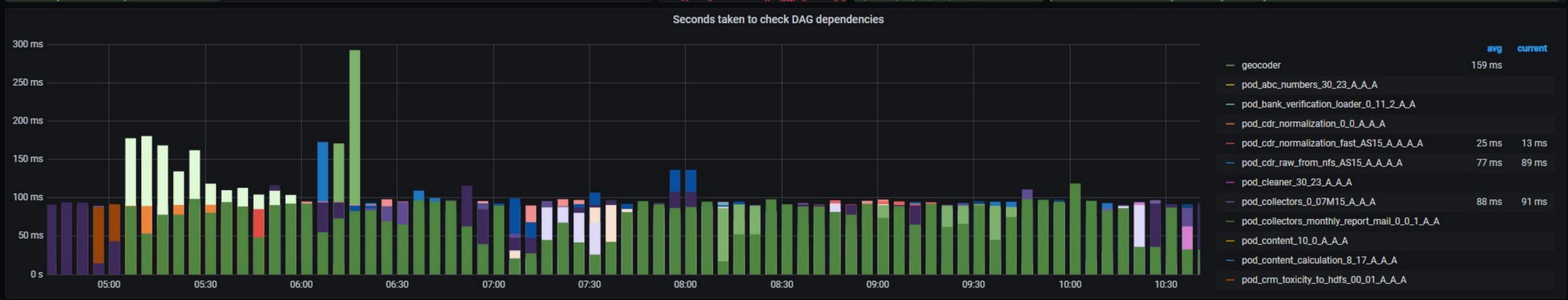
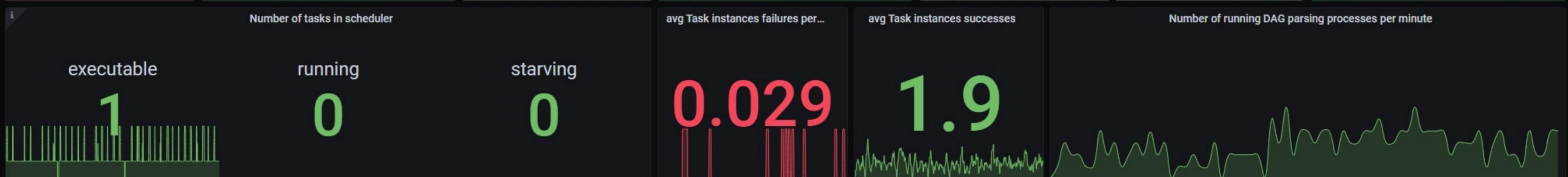
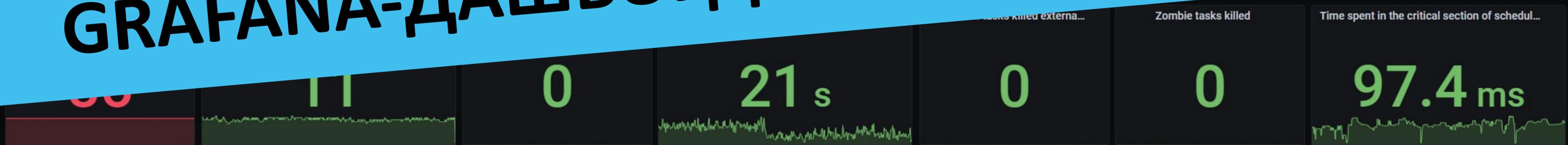
СХЕМА ПОЛУЧЕНИЯ МЕТРИК ИЗ AIRFLOW



ПРОБЛЕМЫ С МОНИТОРИНГОМ



GRAFANA-ДАШБОРД



НА БАЗЕ ЭТОГО РЕАЛИЗОВАНЫ:

01.

**Алертинг по базовым
проблемам**

(Таймауты, ретраи, и т.д.)

02.

**Мониторинг
подвисших процессов**

(При помощи API)

03.

**Конечность времени
жизни инстанса**

ВРЕМЯ ЖИЗНИ СТЕНДА

Стенд жив, пока справедливо следующее:

01.

На стенде в течение последних 25 часов был даг в статусе Running

02.

Стенд был поднят менее чем 7 дней назад

Если стенд не соответствует этим условиям – `namespace` с этим стендом автоматически удаляется.

CRONJOB ДЛЯ УДАЛЕНИЯ СТЕНДА

Workload: airflow-saas-scheduler Active

Namespace: schedulers | Image: artifactory.tdp.corp.tele2.ru:443/airflow-k8s-repo/airflow-saas-scheduler:1.1 | Workload Type: Cron Job

Cron Schedule: */5 * * * *
Every 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 and 55th minute past every hour | Endpoints: n/a | Created: 05/17/2021
Pod Restarts: 0

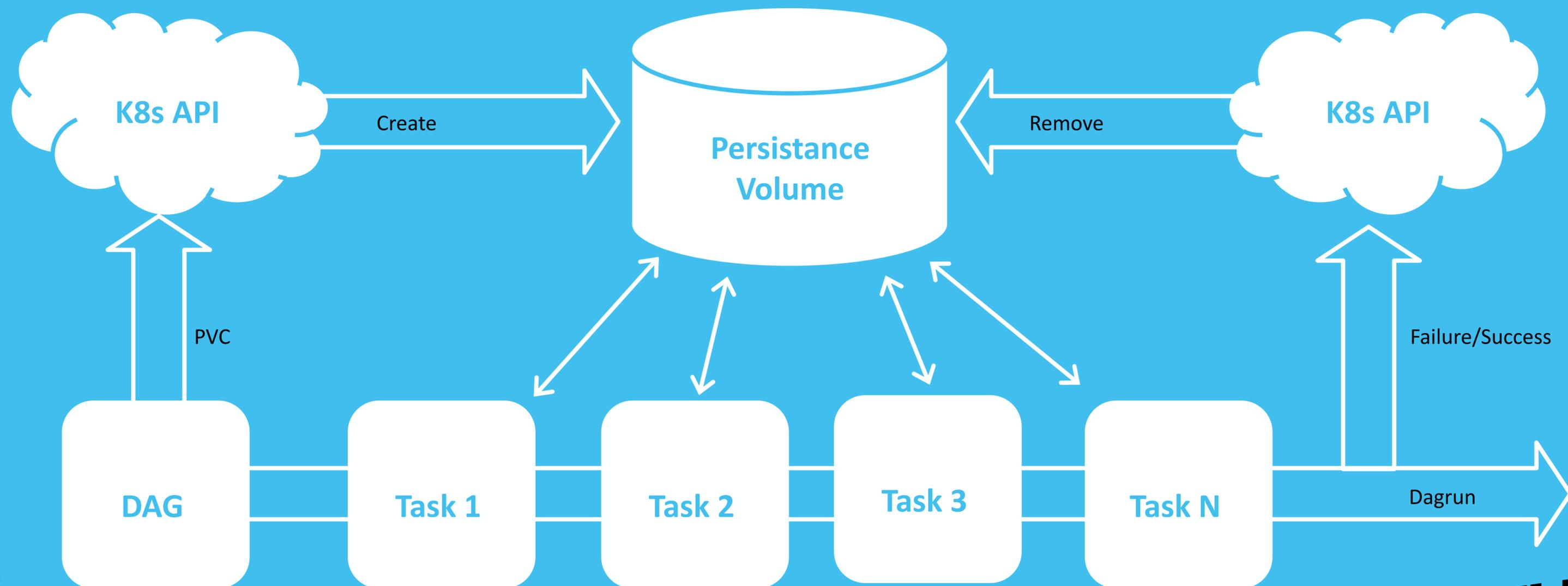
Expand All

▼ Pods
Pods in this workload

Download YAML | Delete

State	Name	Image	Node	
Succeeded	airflow-saas-scheduler-27216745-5pgdd	artifactory.tdp.corp.tele2.ru:443/airflow-k8s-repo/airflow-saas-scheduler:1.1 10.42.5.238 / Created a minute ago / Restarts: 0	t2ru-k8s-04 10.246.11.13	
Succeeded	airflow-saas-scheduler-27216740-lwv9b	artifactory.tdp.corp.tele2.ru:443/airflow-k8s-repo/airflow-saas-scheduler:1.1 10.42.3.95 / Created 6 minutes ago / Restarts: 0	t2ru-k8s-06 10.246.11.15	
Succeeded	airflow-saas-scheduler-27216735-8mgs6	artifactory.tdp.corp.tele2.ru:443/airflow-k8s-repo/airflow-saas-scheduler:1.1 10.42.5.232 / Created 11 minutes ago / Restarts: 0	t2ru-k8s-04 10.246.11.13	

СХЕМА РЕАЛИЗАЦИИ ВРЕМЕННОГО ХРАНИЛИЩА



ПОДВОДНЫЕ
КАМНИ
И ПРОБЛЕМЫ



TELE2

КАК МЫ ПЛАНИРОВАЛИ ВСЕ ЭТО НАЧАТЬ:

01.

**Перенос airflow 1.10.12 в
k8s**

02.

**Адаптация наших
разработок под Airflow 2.0**

03.

Переход на Airflow 2.0

А как все пол



Нет времени объяснять!
Едем сразу в кубер
и на 2.0

Ось?

Капитан!
Вышел Airflow 2.0.1!

Есть, мой капитан!



ДОРАБОТКА K8S OPERATOR

- | Передача конфигурации для Spark-приложений
- | Добавление поддержки временных хранилищ (PVC)

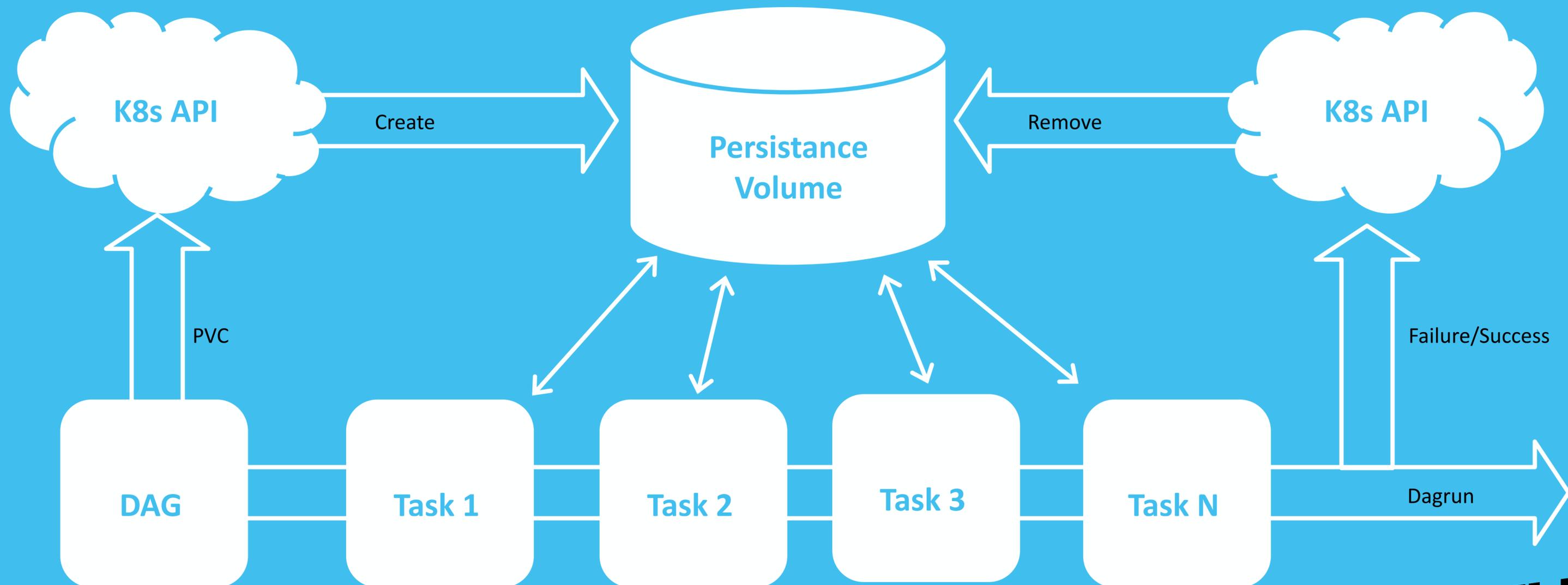
ПРИМЕР ПЕРЕДАЧИ КОНФИГУРАЦИИ

```
abc_regions_loader:
  type: k8s_spark
  driver:
    cores: 1
    memory: 4G
  executor:
    cores: 1
    memory: 4G
    instances: 15
  deploy_mode: cluster
  master: yarn
  java_class: 'com.tele2.tdp.abc.AbcRegionsLoader'
  jars: /extra/*.jar,/etl.jar
  name: abc_regions_loader
  application: /etl.jar
  typesafe_config:
    source:
      rawAbkcPath: "/warehouse/integration/raw_data/{{ds}}"
      regionMincomPath: "/regions_mincom/{{ds}}"
    sink:
      path: "/abc_regions"
  image: artifactory.tdp.corp.tele2.ru/tdp-registry/abc-regions-loader:5-14
```

55

TELE2

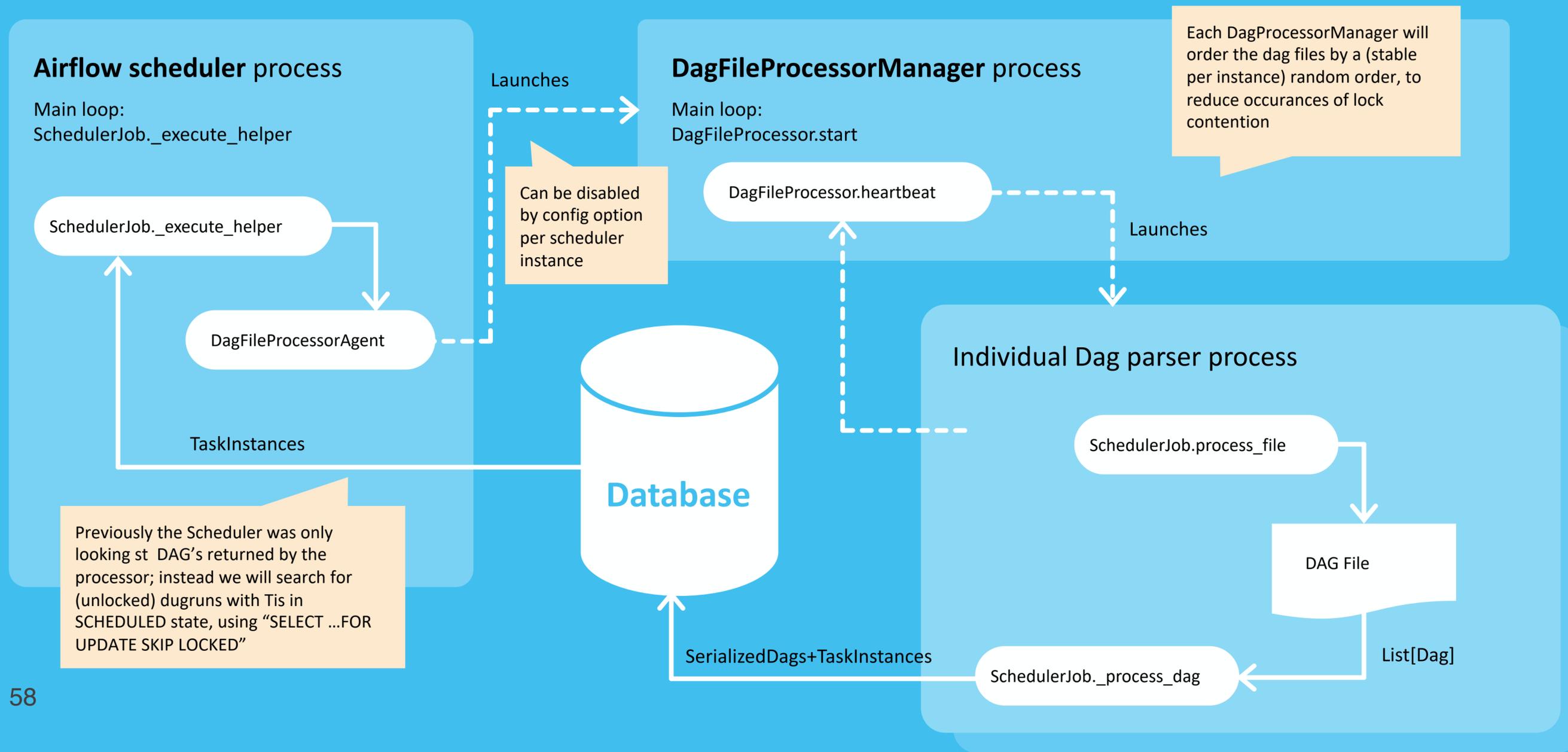
СХЕМА ВЫДЕЛЕНИЯ ВРЕМЕННОГО ХРАНИЛИЩА ЧЕРЕЗ PVC



НО ЧТО-ТО ПОШЛО НЕ ТАК...

<input type="checkbox"/>	Bound	dagrun-shared-pvc-2303cb65a761fc30ba9685eeb54cd1d0	1 GiB	pvc-37701e70-d95a-4f50-aaf5-0d69df0436f0	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-454e4801dc465a89581d8f141f0f1bee	1 GiB	pvc-a2b3fb18-0758-4590-aeff-31f44bdb9d7b	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-4adb0eeef37e1a66a01c36f23a8df0	1 GiB	pvc-b89b9cf5-c6ff-4c15-98db-18f757ec1d41	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-4cbe5c944882ceab38687ae18f80918	1 GiB	pvc-f60ab87f-42b8-400f-884d-2b978e1058de	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-4f847826856baec4ea410bcd21a1719	1 GiB	pvc-6821ccf1-1f7b-4a0c-93ff-75f3ac3221b	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-4f847826856baec4ea410bcd21a1719	1 GiB	pvc-e450f659-1863-4224-b8b0-ab14dc34a39e	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-5316a45dd889244004b14a03099f2cdf	1 GiB	pvc-93dae472-cd4e-4219-a963-4a88c128b97f	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-55ffcd684b63ed0704d9d4e23ea6466	1 GiB	pvc-2faad05c-2f3a-4472-a1d2-ee775f54c1e0	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-66cd0b7e5ee735b83f259619a72d2009	1 GiB	pvc-cb5967b8-ebc3-432b-a587-c9a5ff23d7e8	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-727b7942a27e755f3b338c662bd3e9f	1 GiB	pvc-85ee73cd-5483-4fda-863f-f39c7168ddc5	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-7ac920c19f1cd7fd6034c11a1e97de1b	1 GiB	pvc-404dfc2e-b578-4614-84cf-3690d4d5ccd1	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-7b654586769a009b1a41a3fb3172ac8	1 GiB	pvc-169ae61b-37af-4c91-92a3-7092dc14e7c3	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-7f50c750c15b1f9c30d64cf6722da52d	1 GiB	pvc-d2965085-d46b-4d78-ba0f-90975a573512	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-98dc4b6f611aa313686ed2cb285e74	1 GiB	pvc-c2c5d458-e389-453a-8fa0-1c8b1d847ae3	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-9c6e449f453ddb79f24811e49247db6	1 GiB	pvc-b58bc856-8625-45b7-a24c-f53343527147	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-9ea262d7b8af029324fe2332bc3d6abc	1 GiB	pvc-9e039b09-1254-4ad7-b89a-875c25c3ca8d	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-a0187ad096af54c8b191bc063c5ff1	1 GiB	pvc-96b9f605-35a6-4e25-9602-e7fcc8884878	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-a5d613b100099cd91c6600cf800c69e6	1 GiB	pvc-d3dc342b-8d22-48b8-b0fc-c4df5c1e75fc	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-a6d12578ea8cf63aae38709a39974ff9	1 GiB	pvc-0cbbfb47-092f-4a6c-bab0-dc93b65bc71c	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-b754c5d52e19796075b2759c7d10b1a	1 GiB	pvc-70e63cb5-8243-4758-8118-18749bb1d435	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-be8d7617ea6ec5b2209a0c31c1c95107	1 GiB	pvc-6a14a5ab-80a8-4c98-9f99-9f2ed3e5f8dd	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-c7f2e4a9164a35bbe6419a307321162c	1 GiB	pvc-7316fa65-clab-4f24-9681-ea75009b03cc	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-cf9c17b81ad7a61feb017d3aee4a9b9	1 GiB	pvc-a13c0c38-88c0-4041-b248-702cdc10ebfd	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-d9f04fd223bde76ee750b011ed5d1a7d	1 GiB	pvc-9500a792-b3d1-46fc-a0c3-3b6835f27b18	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-dc4e5866cd524c336977fb0357905a7e	1 GiB	pvc-b5765548-c22d-4c77-ace5-5522850b9b1f	ceph-csi	
57	Bound	dagrun-shared-pvc-dd5fcc040267052cf6a3e4a33e9ddc0a	1 GiB	pvc-021c956-5018-483e-8cac-370e30438e2d	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-e3210d1941c848562fb00f79181abc25	1 GiB	pvc-95c54cec-dc7a-4936-8ded-bdf003040cf9	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-e80373e4c8afca2f32c22e7c964ad1eb	1 GiB	pvc-3702efe3-5d32-4ac9-a8c5-902840cb1019	ceph-csi	
<input type="checkbox"/>	Bound	dagrun-shared-pvc-e8e27e734083b8b9ea5152c0fd3efdc	1 GiB	pvc-df33b100-1026-4681-bb09-a5e2549da708	ceph-csi	

ПРОЦЕСС ПАРСИНГА DAG'ОВ



ПРИЧИНА ПРОБЛЕМЫ

```
class DagFileProcessorManager(LoggingMixin):  
    def __init__(  
        self,  
        dag_directory: Union[str, "pathlib.Path"],  
        max_runs: int,  
        ...  
    ):  
        super().__init__()  
        self._file_paths: List[str] = []  
        ...  
        # Map from file path to the processor  
        self._processors: Dict[str, AbstractDagFileProcessorProcess] = {}  
        ...  
  
    def _refresh_dag_dir(self):  
        now = timezone.utcnow()  
        elapsed_time_since_refresh = (now - self.last_dag_dir_refresh_time).total_seconds()  
        if elapsed_time_since_refresh > self.dag_dir_list_interval:  
            ...  
            self._file_paths = list_py_file_paths(self._dag_directory)  
            self.last_dag_dir_refresh_time = now  
            self.set_file_paths(self._file_paths)  
            ...
```

1

```
def set_file_paths(self, new_file_paths):  
    self._file_paths = new_file_paths  
    self._file_path_queue = [x for x in self._file_path_queue if x in new_file_paths]  
    # Stop processors that are working on deleted files  
    filtered_processors = {}  
    for file_path, processor in self._processors.items():  
        if file_path in new_file_paths:  
            filtered_processors[file_path] = processor  
        else:  
            processor.terminate()  
            self._file_stats.pop(file_path)  
    self._processors = filtered_processors
```

2

А ТАКЖЕ — ПРОБЛЕМЫ С ТАЙМАУТАМИ

```
File Path          PID  Runtime    # DAGs    # Errors  Last Runtime  Last Run
-----
/opt/airflow/dags/airflow_dags.zip 20915 121.28s    0         0
===== - None
[2021-07-19 15:41:28,802] {dag_processing.py:1164} ERROR - Processor for /opt/airflow/dags/airflow_dags.zip with PID 209
15 started at 2021-07-19T15:39:27.523653+00:00 has timed out, killing it. - None
[2021-07-19 15:41:29,934] {dag_processing.py:604} INFO - Processing files using up to 2 processes at a time - None
[2021-07-19 15:41:29,934] {dag_processing.py:605} INFO - Process each file at most once every 30 seconds - None
[2021-07-19 15:41:29,935] {dag_processing.py:607} INFO - Checking for new files in /opt/airflow/dags every 60 seconds -
None
[2021-07-19 15:41:29,935] {dag_processing.py:755} INFO - Searching for files in /opt/airflow/dags - None
[2021-07-19 15:41:29,961] {dag_processing.py:758} INFO - There are 1 files in /opt/airflow/dags - None
[2021-07-19 15:41:31,041] {dag_processing.py:1123} INFO - Finding 'running' jobs without a recent heartbeat - None
[2021-07-19 15:41:31,043] {dag_processing.py:1127} INFO - Failing jobs without heartbeat after 2021-07-19 15:36:31.04354
8+00:00 - None
```

РЕШЕНИЕ —
ГЕНЕРАТОР В ГЕНЕРАТОРЕ

Я слышал
ты любишь
генерировать?

Мы добавили еще один генератор, чтобы ты
мог генерировать генераторы, которые
генерируют даги



РЕЗУЛЬТАТ

```
File Path                                     PID  Runtime   # DAGs   # Errors  Last Runtime  Last Run
-----
/opt/airflow/dags/current/pod_runner_tele2_dag1          1      1.22s     1         0    1.22s      2021-09-30T13:17:20
/opt/airflow/dags/current/pod_runner_tele2_dag2          1      1.25s     1         0    1.25s      2021-09-30T13:17:31
/opt/airflow/dags/current/pod_runner_tele2_dag3          1      1.17s     1         0    1.17s      2021-09-30T13:17:08
/opt/airflow/dags/current/pod_runner_tele2_dag4          1      1.13s     1         0    1.13s      2021-09-30T13:17:08
/opt/airflow/dags/current/pod_runner_tele2_dag5          1      1.35s     1         0    1.35s      2021-09-30T13:17:21
/opt/airflow/dags/current/pod_runner_tele2_dag6          1      1.17s     1         0    1.17s      2021-09-30T13:17:33
...
/opt/airflow/dags/current/pod_runner_tele2_dag88         1      2.46s     1         0    2.46s      2021-09-30T13:17:10
=====
[2021-09-30 13:17:41,800] {dag_processing.py:1128} INFO - Finding 'running' jobs without a recent heartbeat
[2021-09-30 13:17:41,800] {dag_processing.py:1132} INFO - Failing jobs without heartbeat after 2021-09-30 13:12:41.800954+00:00
[2021-09-30 13:17:52,306] {dag_processing.py:1128} INFO - Finding 'running' jobs without a recent heartbeat
[2021-09-30 13:17:52,307] {dag_processing.py:1132} INFO - Failing jobs without heartbeat after 2021-09-30 13:12:52.307517+00:00
```

А ТЕПЕРЬ —

Демонстрация!

TELE2

РЕЗУЛЬТАТЫ ПРОЕКТА



TELE2

ЧЕГО МЫ ДОБИЛИСЬ

**Централизованное
хранилище логов**

**K8s-executor
сократил затраты
ресурсов**

**Ресурсы больше
не простаивают**

**Версионированность
Connections и Variables**

**Полная изоляция
пользователей**

Администрирование

**Пользователь сам
выбирает версию Airflow и
набор библиотек**

**Масштабируемость
проекта ограничена
лишь ресурсами**

**Отказоустойчивость
Production**

ИТОГИ В ЦИФРАХ



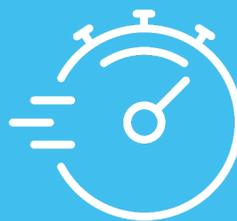
1,5
минуты

Время подготовки стенда **1,5**
минуты вместо часа



16 / 30
vCPU / Gb RAM

Преаллокация ресурсов сократилась
с 48 vCPU и 180Gb RAM до 16vCPU и 30Gb RAM



-72
минуты

Time-to-market в среднем
сократился на **72** минуты



Процесс тестирования новых версий и
компонентов Airflow теперь занимает
дни, а не недели.

Вопросы?

TELE2

**СПАСИБО
ЗА ВНИМАНИЕ**

TELE2