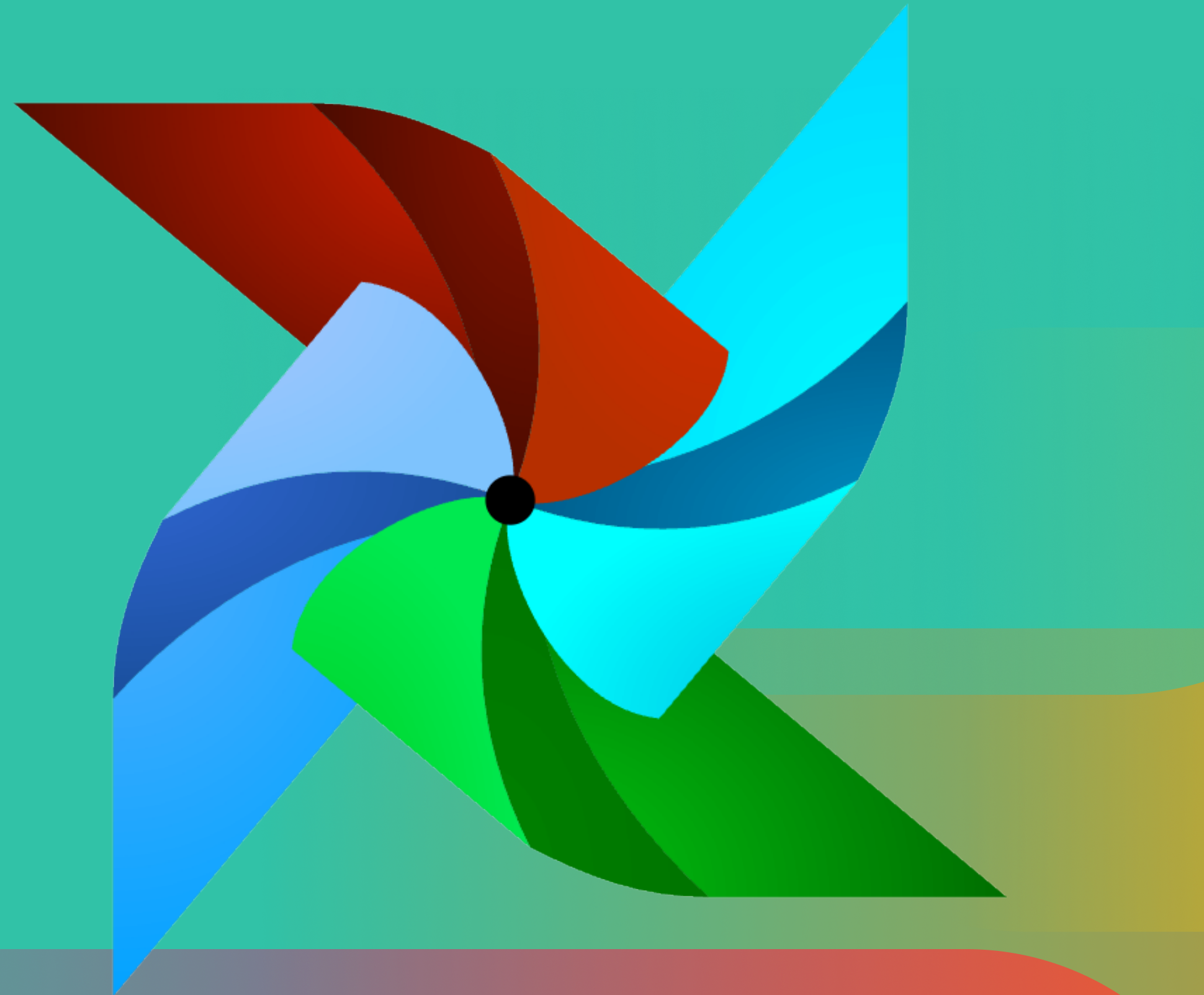


# Airflow в кровавом энтерпрайзе



обо мне



Илья Савин

Инженер ИТ инфраструктуры



# Что такое Apache Airflow

Это открытое программное обеспечение для создания, выполнения, мониторинга и оркестровки потоков операций по обработке данных. (с) wikipedia

# ОСНОВНЫЕ КОМПОНЕНТЫ apache airflow

Scheduler

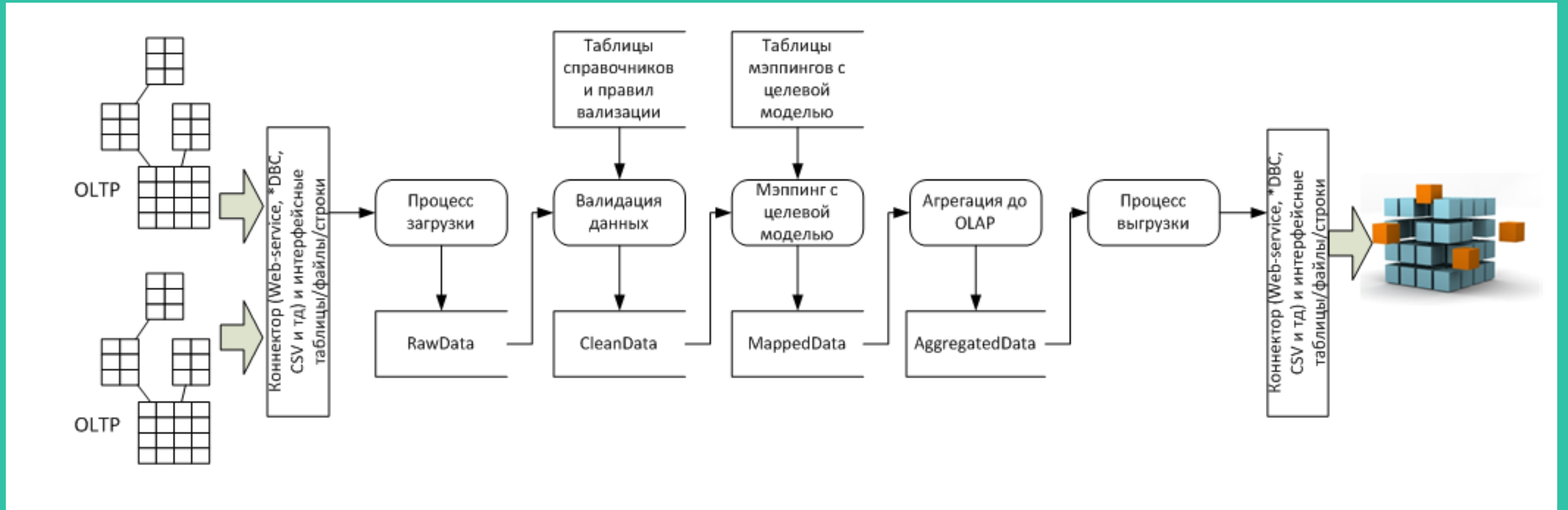
Triggerer

Worker

Web-server

# Что такое ETL

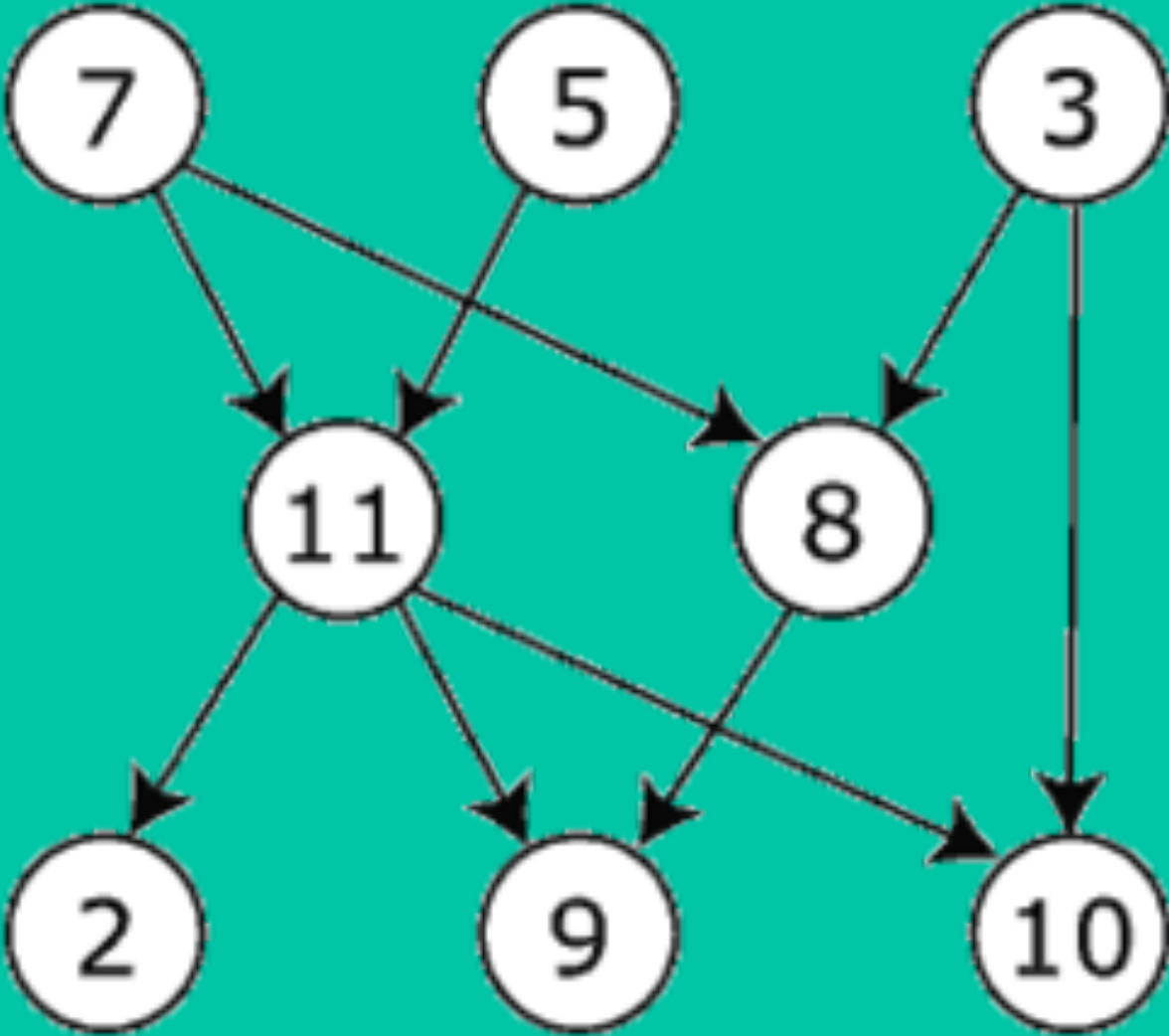
Extract, Transform, Load



Красиво нарисованный процесс ETL из открытого источника (гугл поиск по картинкам)

# Что такое DAG

Directed Acyclic Graph



Направленный ациклический граф (с) википедия

# ОБЩИЙ ВИД ИНТЕРФЕЙСА AIRFLOW

The screenshot displays the Apache Airflow web interface. At the top, there is a navigation bar with the Airflow logo, menu items (DAGs, Security, Browse, Admin, Docs), the current time (21:11 UTC), and a user profile (RH). Below the navigation bar, the 'DAGs' section is visible, featuring a filter for 'All 26' DAGs, with sub-filters for 'Active 10' and 'Paused 16'. A search bar and a 'Filter DAGs by tag' input are also present. The main content is a table listing various DAGs with columns for DAG name, Owner, Runs, Schedule, Last Run, Recent Tasks, Actions, and Links. Each DAG row includes a toggle switch, a status indicator (green or grey circle), and a series of task status icons (circles) representing the DAG's execution history.

DAG	Owner	Runs	Schedule	Last Run	Recent Tasks	Actions	Links
<input checked="" type="checkbox"/> example_bash_operator example example2	airflow	2	0 0 ***	2020-10-26, 21:08:11	6	[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_branch_dop_operator_v3 example	airflow		* / 1 * * * *			[▶] [↺] [🗑️]	...
<input type="checkbox"/> example_branch_operator example example2	airflow	1	@daily	2020-10-23, 14:09:17	11	[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_complex example example2 example3	airflow	1 1	None	2020-10-26, 21:08:04	37 37	[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_external_task_marker_child	airflow	1	None	2020-10-26, 21:07:33	2	[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_external_task_marker_parent	airflow	1	None	2020-10-26, 21:08:34	1	[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_kubernetes_executor example example2	airflow		None			[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_kubernetes_executor_config example3	airflow	1	None	2020-10-26, 21:07:40	5	[▶] [↺] [🗑️]	...
<input checked="" type="checkbox"/> example_nested_branch_dag example	airflow	1	@daily	2020-10-26, 21:07:37	9	[▶] [↺] [🗑️]	...
<input type="checkbox"/> example_passing_params_via_test_command example	airflow		* / 1 * * * *			[▶] [↺] [🗑️]	...

- СПИСОК ДАГОВ
- ОПИСАНИЕ ДОГОВ
- СТАТУС ТАСОК

# grid view

отображение по времени  
исполнения и статус  
конкретных тасок во всем даге

The screenshot displays the Airflow web interface for a DAG named 'example\_bash\_operator'. The interface is organized into several sections:

- Navigation Bar:** Includes links for DAGs, Datasets, Security, Browse, Admin, and Docs. The current time is 08:17 UTC.
- DAG Header:** Shows the DAG name 'example\_bash\_operator', its schedule '0 0 \*\*\*', and the next run time '2023-05-27, 00:00:00'.
- View Options:** A row of icons for different views: Grid (selected), Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, and Details. Below these are links for 'Code' and 'Audit Log'.
- Filters and Controls:** A section for filtering runs by date (27-05-2023 08:13:44), page size (25), run types, and run states. It includes a 'Clear Filters' button and an 'Auto-refresh' toggle.
- Legend:** A row of colored boxes representing different run states: deferred, failed, queued, removed, restarting, running, scheduled, shutdown, skipped, success, up\_for\_reschedule, up\_for\_retry, upstream\_failed, and no\_status.
- Grid View:** A table showing the execution of tasks across three DAG runs. The tasks listed are runme\_0, runme\_1, runme\_2, also\_run\_this, this\_will\_skip, run\_after\_loop, and run\_this\_last. The 'Duration' column shows the execution time for each task, and the 'Status' column uses colored squares to indicate the state of each task.
- Summary Panel:** A panel on the right side of the grid view providing a 'DAG Runs Summary' and a 'DAG Summary'. The 'DAG Runs Summary' includes metrics like 'Total Runs Displayed' (3), 'Total success' (3), 'First Run Start' (2023-05-27, 07:53:57 UTC), 'Last Run Start' (2023-05-27, 07:54:11 UTC), 'Max Run Duration' (00:00:06), 'Mean Run Duration' (00:00:05), and 'Min Run Duration' (00:00:05). The 'DAG Summary' shows 'Total Tasks' (7), 'BashOperators' (6), and 'EmptyOperator' (1).

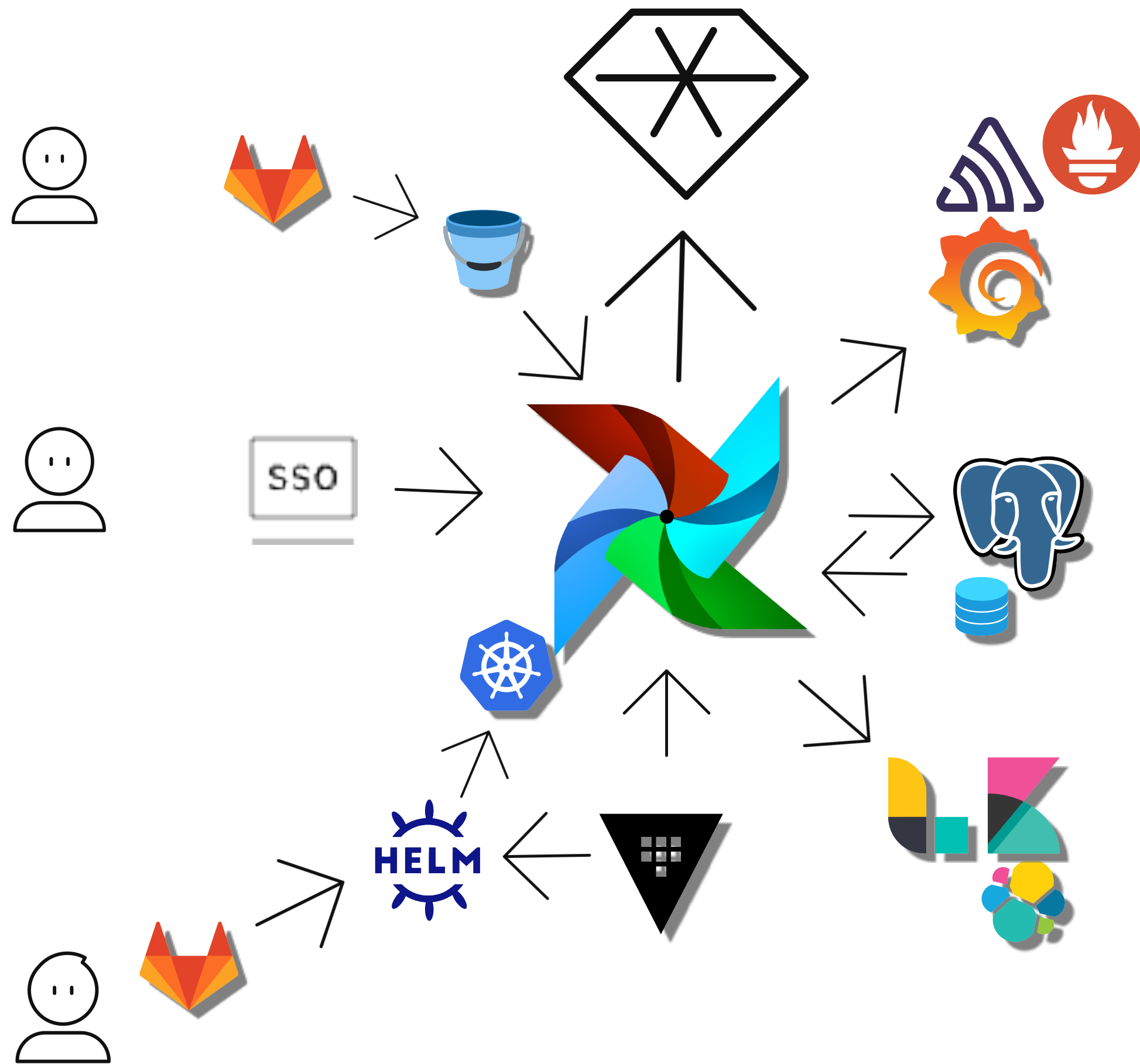


# graph view

The screenshot shows the Airflow web interface for a DAG named 'example\_bash\_operator'. The top navigation bar includes 'Airflow', 'DAGs', 'Security', 'Browse', 'Admin', and 'Docs'. The current time is 10:46 PDT (-07:00) and the user is 'FB'. The DAG status is 'success' with a schedule of '0 0 \* \* \*'. Below the DAG name, there are tabs for 'Tree', 'Graph' (selected), 'Calendar', 'Task Duration', 'Task Tries', 'Landing Times', 'Gantt', 'Details', and 'Code'. A toolbar contains icons for play, refresh, and delete. The main area shows a filter for '2021-06-02T09:27:27-C', 'Runs: 25', and a 'Run' dropdown set to 'manual\_2021-06-02T16:27:26.797940+00:00'. There is an 'Update' button and a 'Find Task...' search box. Below the filters, there are buttons for 'BashOperator' and 'DummyOperator', and a legend for task statuses: 'queued', 'running', 'success', 'failed', 'up\_for\_retry', 'up\_for\_reschedule', 'upstream\_failed', 'skipped', 'scheduled', and 'no\_status'. An 'Auto-refresh' toggle is also present. The graph view displays a DAG with the following tasks and dependencies:

```
graph LR; runme_0[runme_0] --> also_run_this[also_run_this]; runme_1[runme_1] --> run_after_loop[run_after_loop]; runme_2[runme_2] --> this_will_skip[this_will_skip]; also_run_this --> run_after_loop; also_run_this --> run_this_last[run_this_last]; run_after_loop --> run_this_last; this_will_skip --> run_this_last;
```

отображение  
зависимостей  
тасок в даге



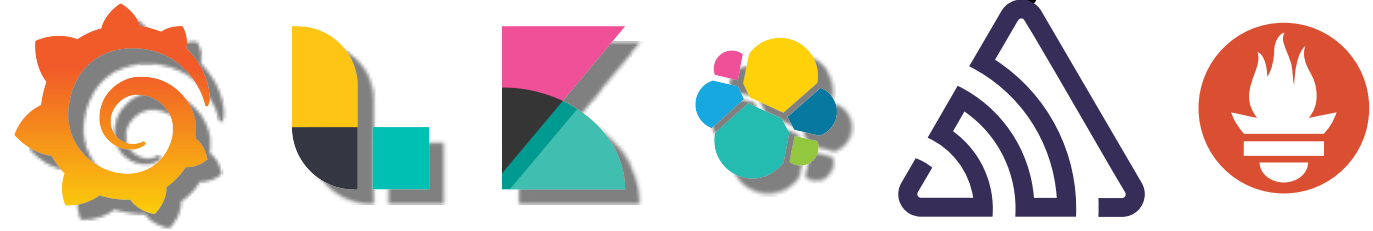
## Плюсы

- + Airflow запускается предсказуемо и масштабируемо (kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продом.
- + Подключен secret backend для connections и секретов конфигурации(Hashicorp Vault).
- + Отдельно вынесенный кластер БД с метаданными(postgres).
- + Вход по SSO(ADFS).
- + Подключен log backend(elastic).
- + Выводятся метрики + алертинг (statsd+GrafanaOnCal).

## Минусы

- Сложное добавление коннекшенов\переменных.
- Увеличение времени между созданием дага и запуском в прод.

Observability



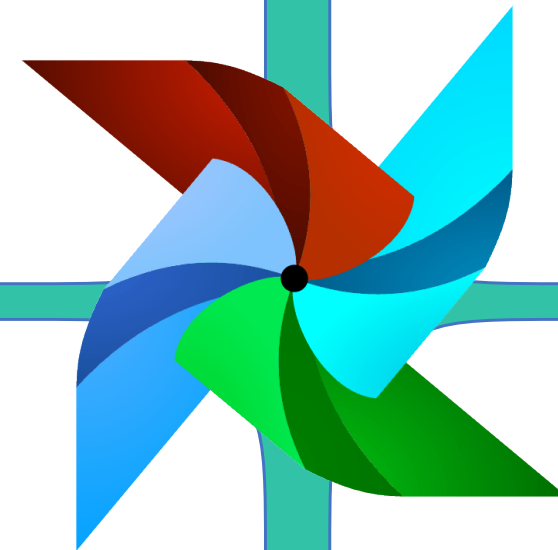
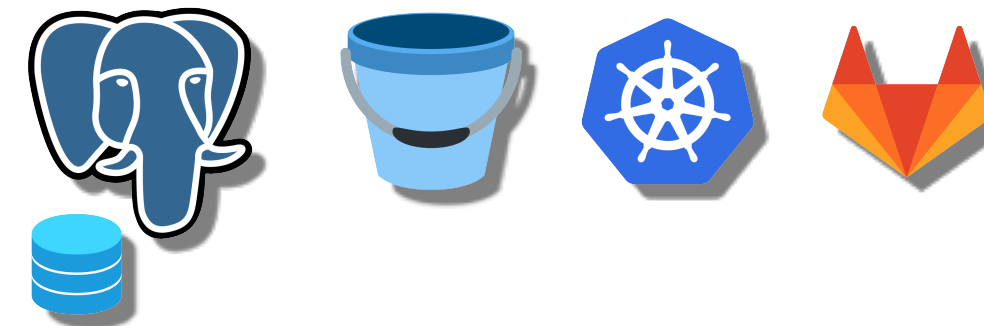
Security



VCS



High availability



## Плюсы

Быстрое восстановление  
Масштабируемость  
Отказоустойчивость

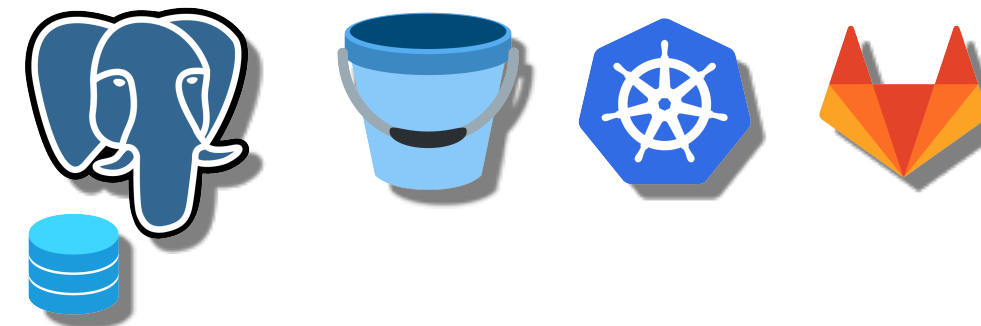
## Минусы

Увеличение времени доставки кода

## Описание

Постгрес - хранение метаданных  
s3 bucket - хранение логов и кода  
k8s - инфраструктура для запуска приложения  
gitlab - быстрое восстановления работоспособного состояния приложения и кода при сбоях.

## High availability



## Плюсы

Код DAGов и приложения версионруется, проверяется, аппрувается, можно спокойно вести разработку без боязни положить прод, так как есть ci\cd и пачка сред

## Минусы

Код медленнее попадает на прод, усложняется поддержка приложения

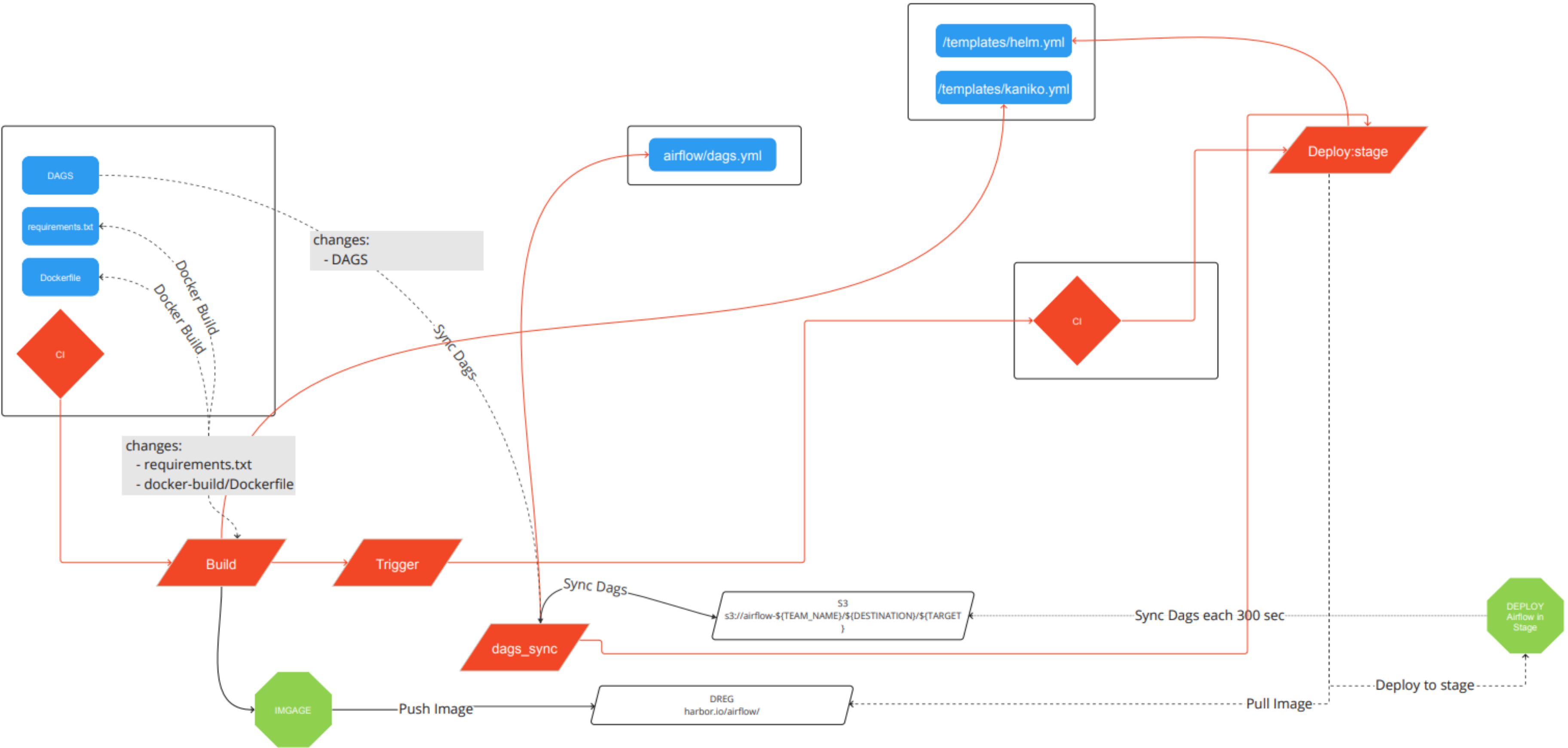
## VCS



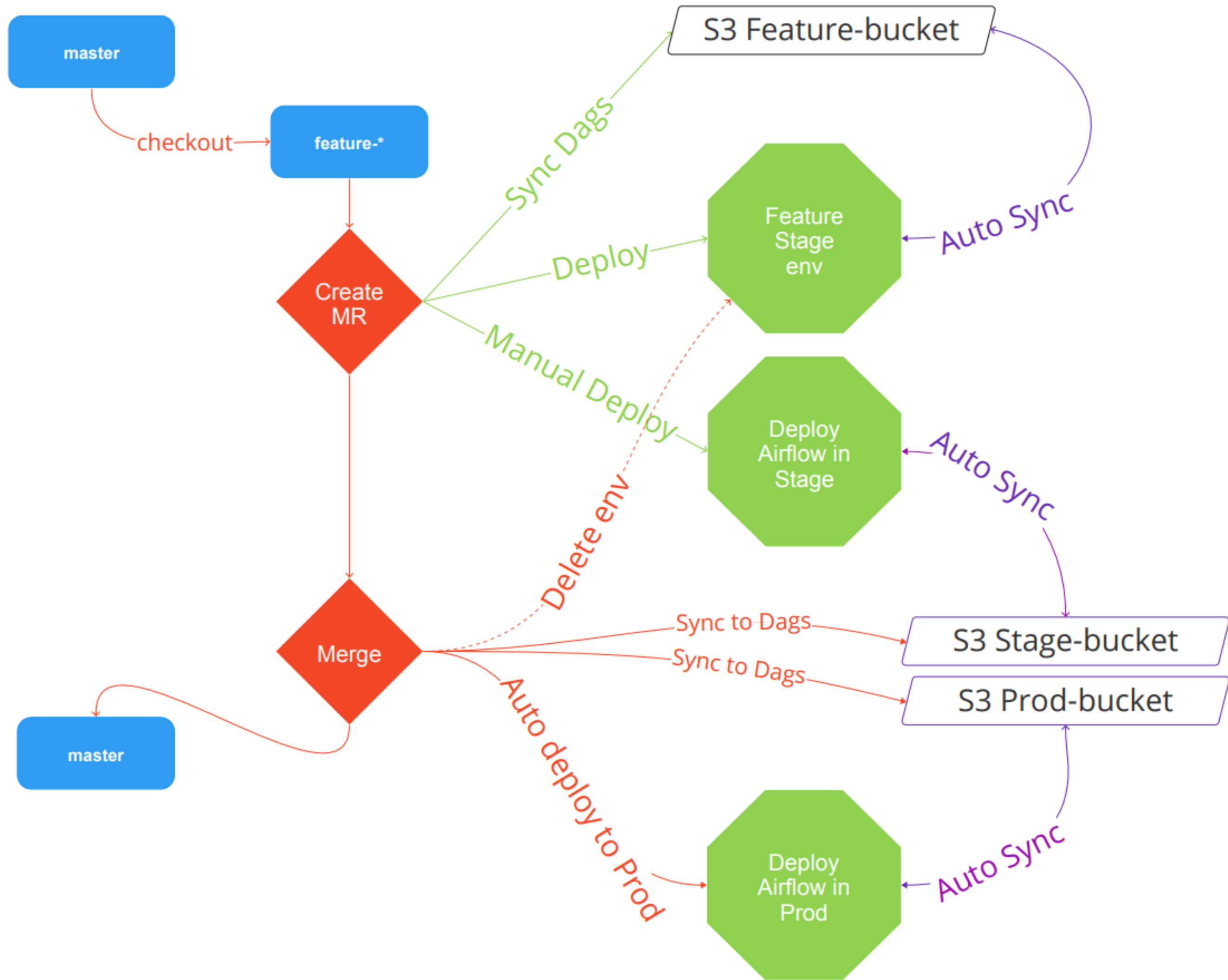
## Описание

Gitlab - ci\cd+git  
Helm - темплейтирование приложения

# На примере Stage ветки



changes:  
- requirements.txt  
- docker-build/Dockerfile





Observability

Описание

StatsD - экспорт метрик

Elastic Stack - сбор и хранение логов + поиск

Sentry - сбор и визуализация трейсов

Grafana - визуализация метрик

Плюсы

Айрфлоу отдает метрики и логи в предназначенные для этого системы, взамен мы получаем наблюдаемость и вдумчивый дебаг

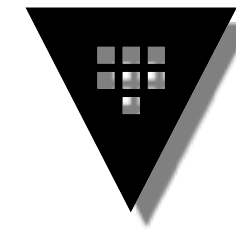
Минусы

Поддержка этих сервисов требует внимания.



## Описание

Hashicorp Vault - хранение и создание секретов  
SSO - доступ по LDAP  
Gitlab - хранение CI переменных, ограничение доступа к CI и репозитория по группам



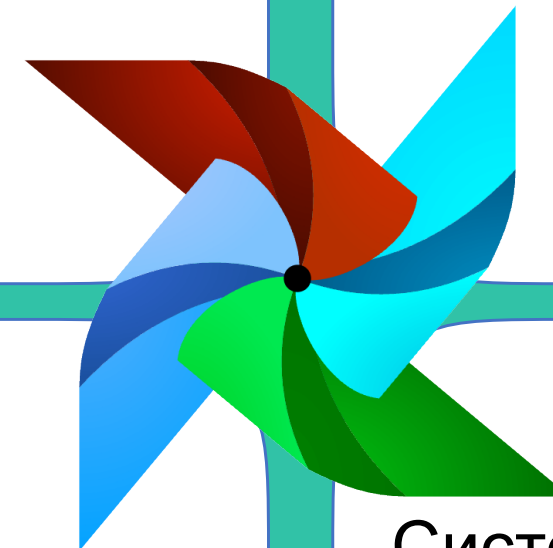
Security

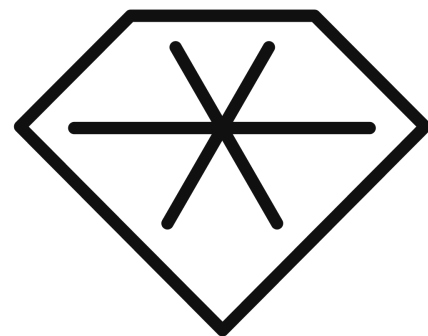
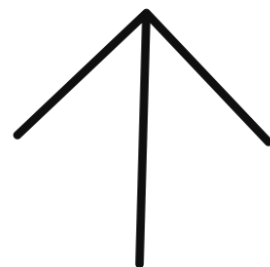
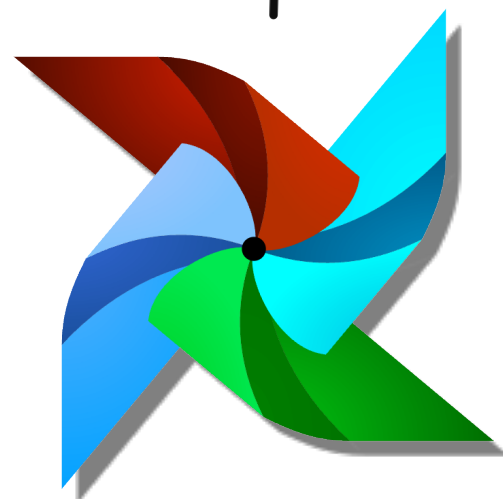
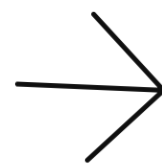
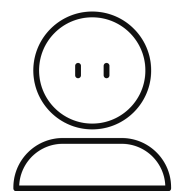
## Плюсы

Секреты версионированы и отслеживаются, появляется возможность автоматического обновления секретов.  
Убирается ручное управление пользователями и их группами

## Минусы

Системы требуют поддержки.  
Медленное изменение коннекшенов по сравнению с UI airflow



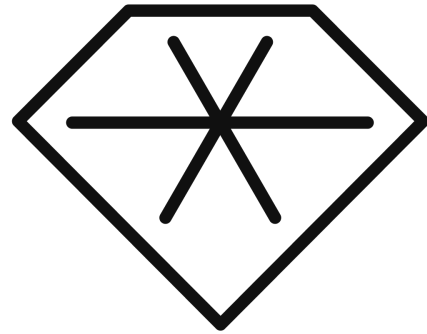
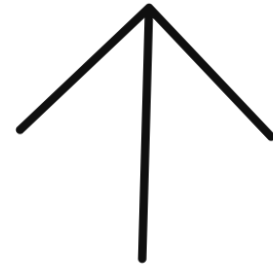
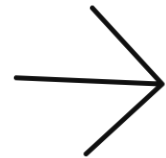
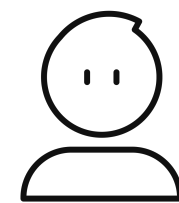
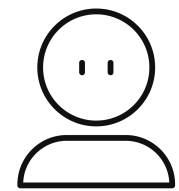


## Плюсы

- + Airflow работает, ETL процесс запущены
- + Быстрая выкатка дагов на прод
- + Легкое добавление секретов и переменных

## Минусы

- Пользователи запускают Airflow локально
- Окружение отличается от локалки к локалке
- Немасштабируемо
- Даги не ревьюятся, хранятся в “надежном” месте
- Секреты хранятся в “надежном” месте и их знает немного ограниченный круг лиц
- Логи локальные
- Метрики локальные
- БД с метаданными локальная

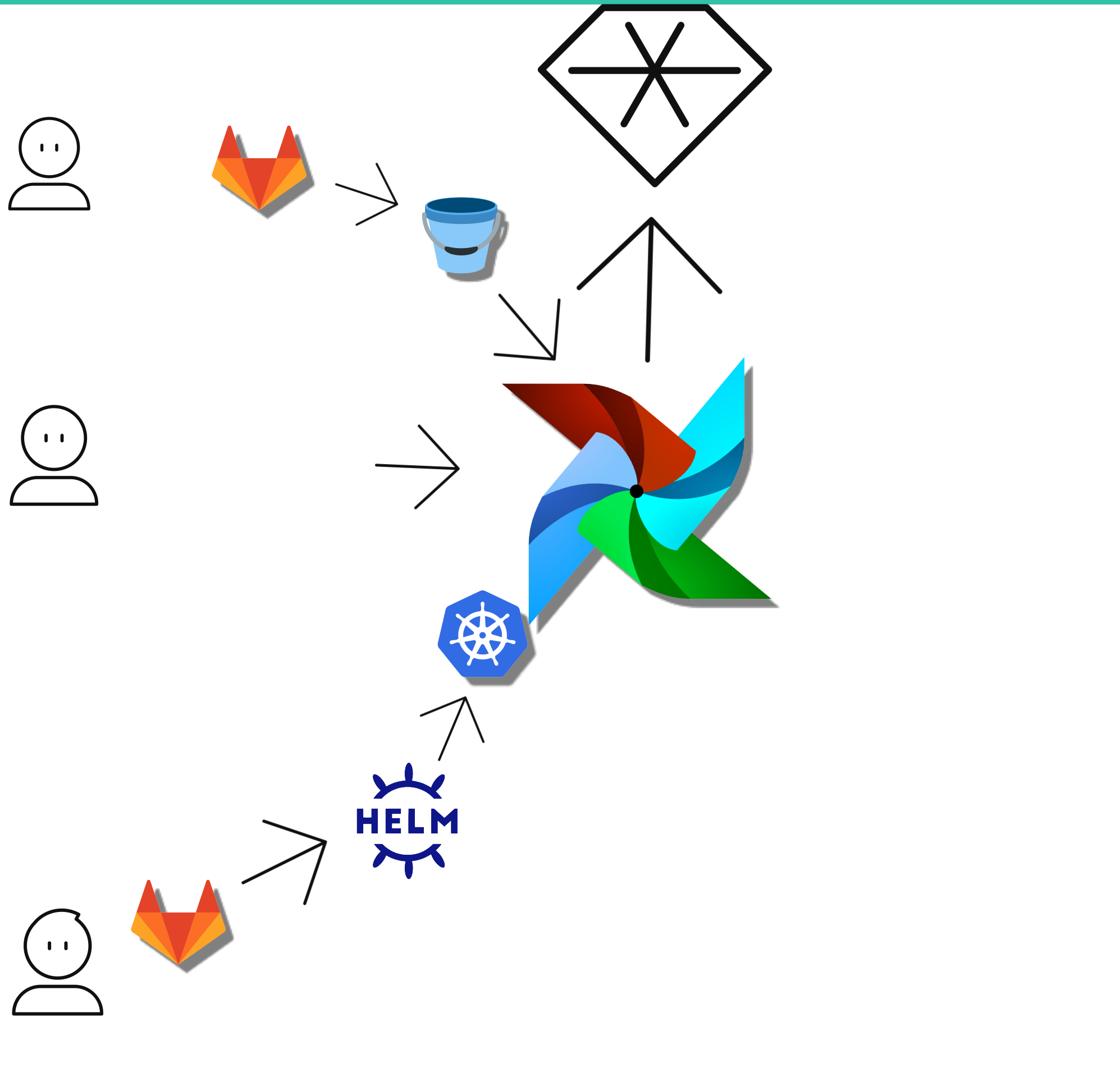


## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm)
- + Быстрая выкатка дагов на прод
- + Легкое добавление секретов и переменных

## Минусы

- Даги не ревьюятся, хранятся в “надежном” месте
- Секреты хранятся в “надежном” месте
- Логи локальные
- Метрики локальные.
- БД с метаданными локальная

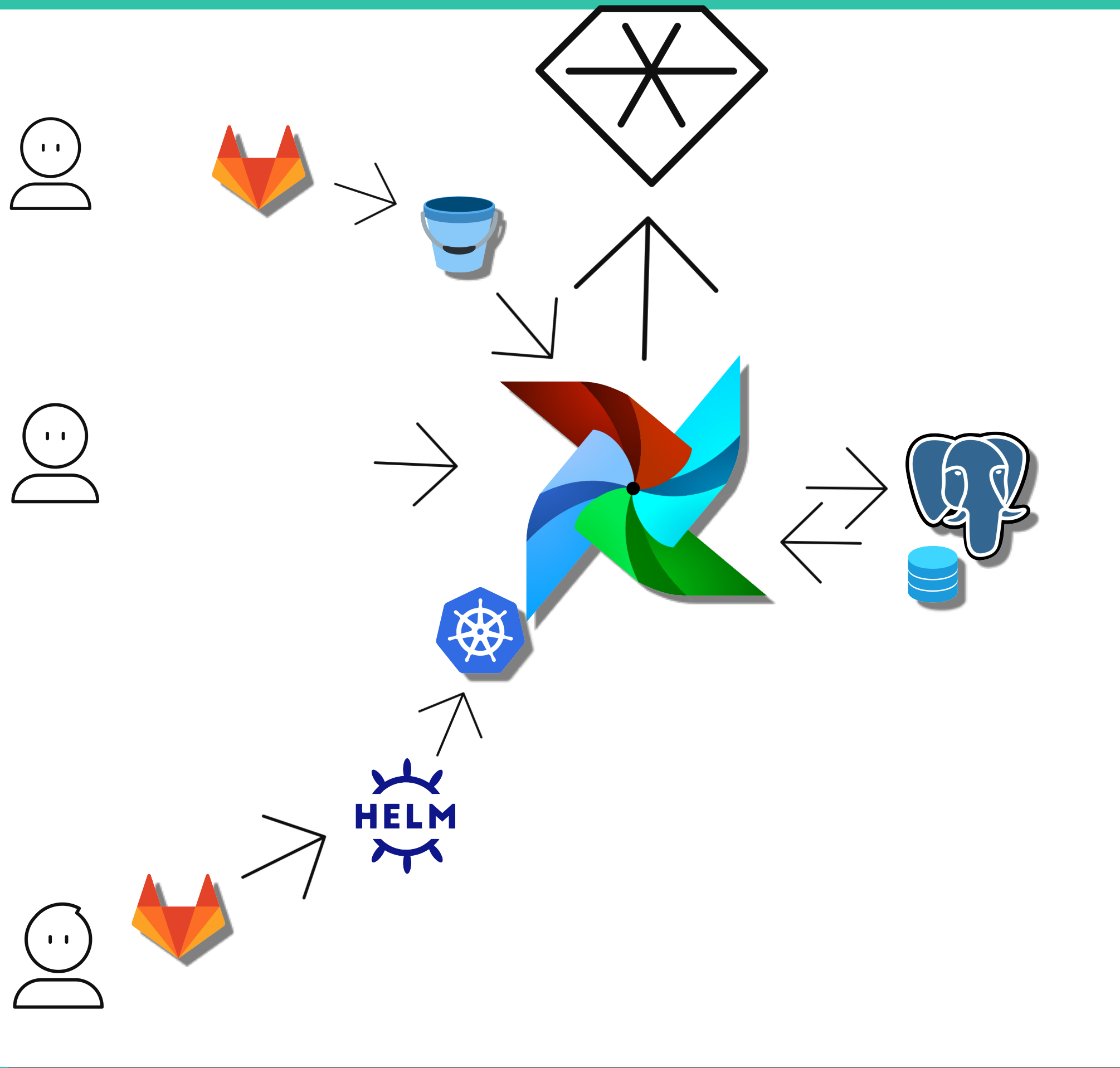


## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продам
- + Легкое добавление секретов и переменных

## Минусы

- Увеличение времени между созданием дага и запуском в прод
- Ненадежно
- Секреты хранятся в “надежном” месте
- Логи локальные
- Метрики локальные.
- БД с метаданными локальная

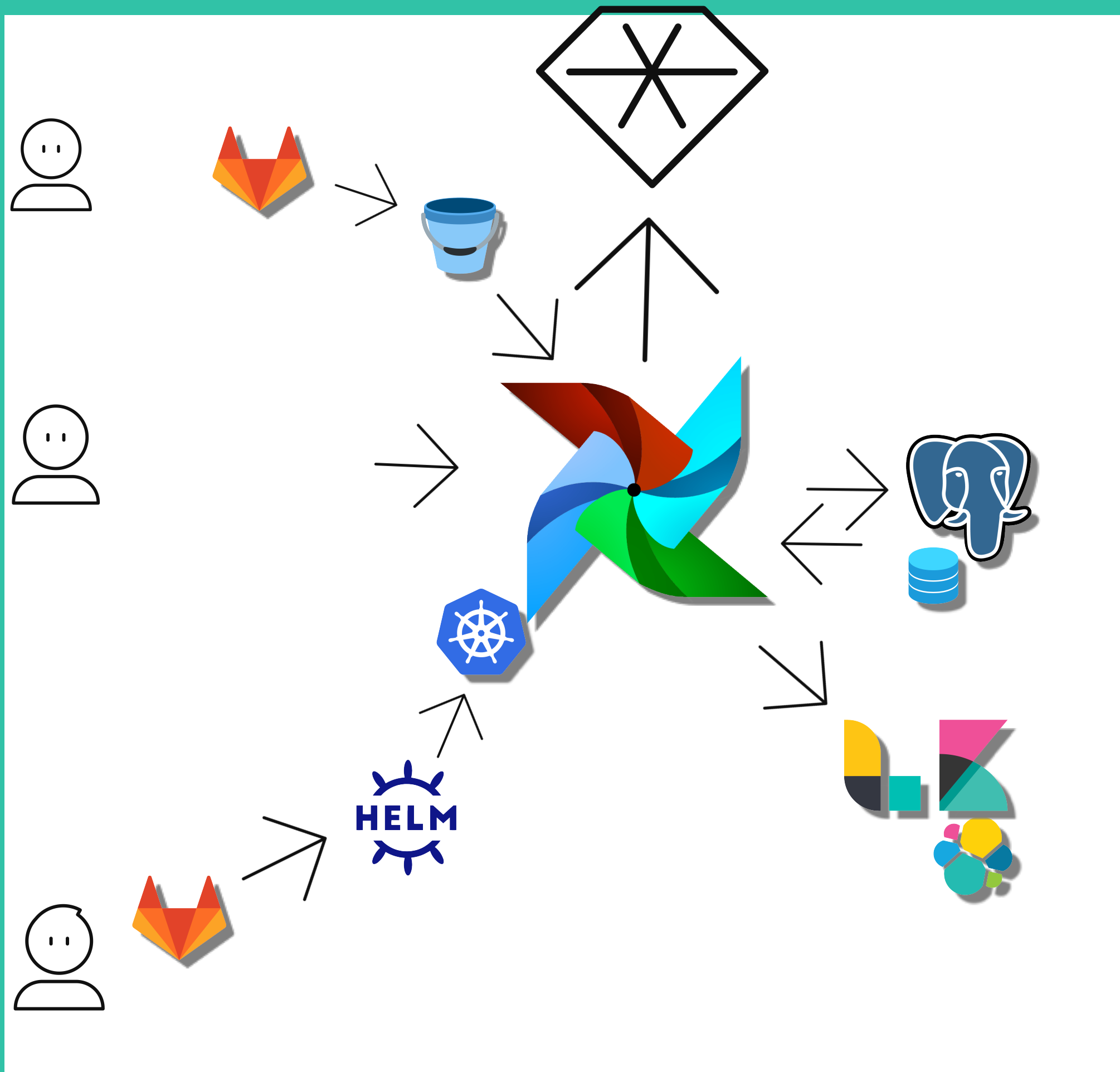


## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продам
- + Легкое добавление секретов и переменных
- + Отдельно вынесенный кластер БД с метаданными(postgres)

## Минусы

- Увеличение времени между созданием дага и запуском в прод
- Ненадежно
- Секреты хранятся в “надежном” месте
- Логи локальные
- Метрики локальные

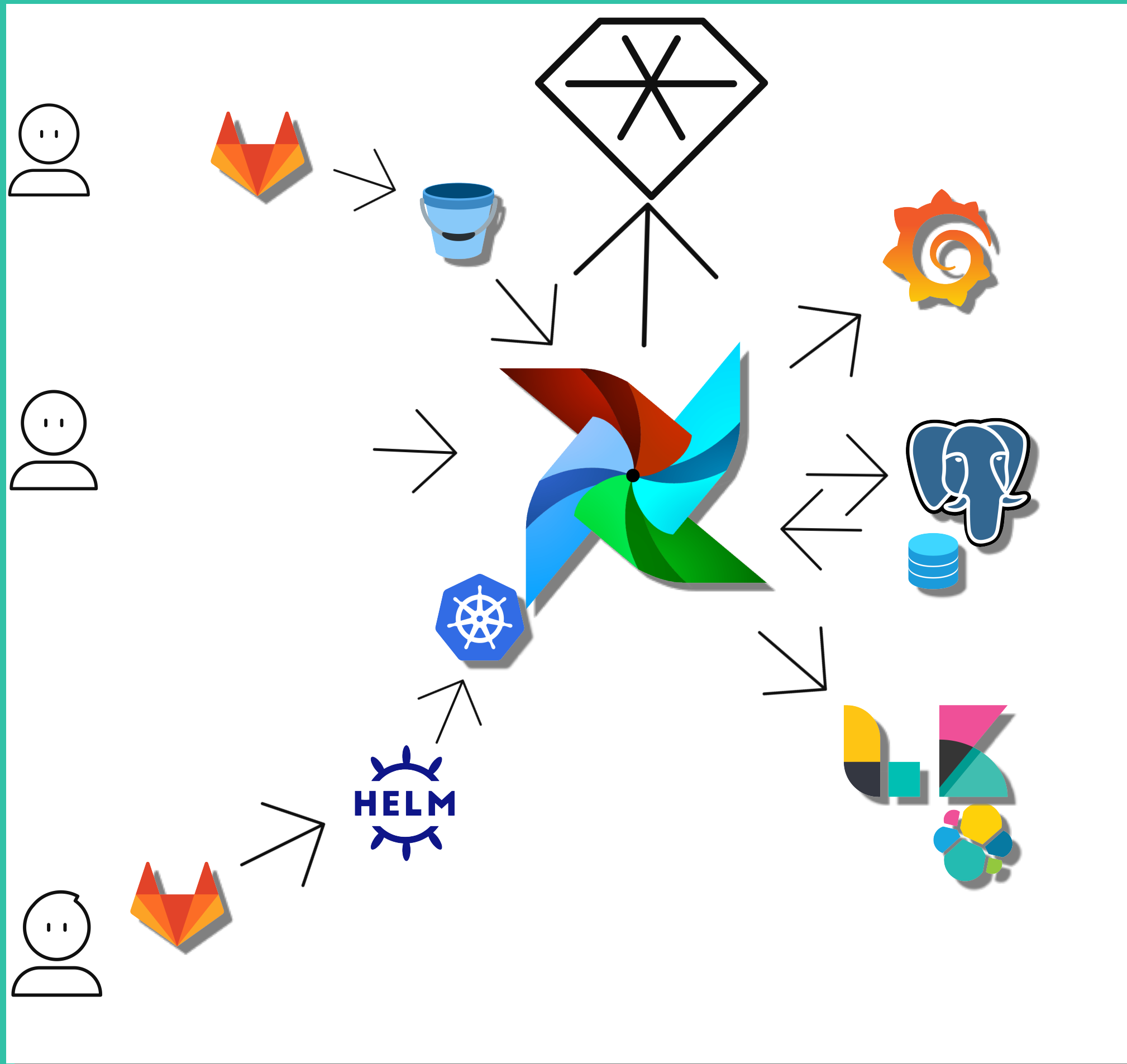


## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продам
- + Легкое добавление секретов и переменных
- + Отдельно вынесенный кластер БД с метаданными(postgres)
- + Подключен log backend(elastic)

## Минусы

- Увеличение времени между созданием дага и запуском в прод
- Секреты хранятся в надежном месте

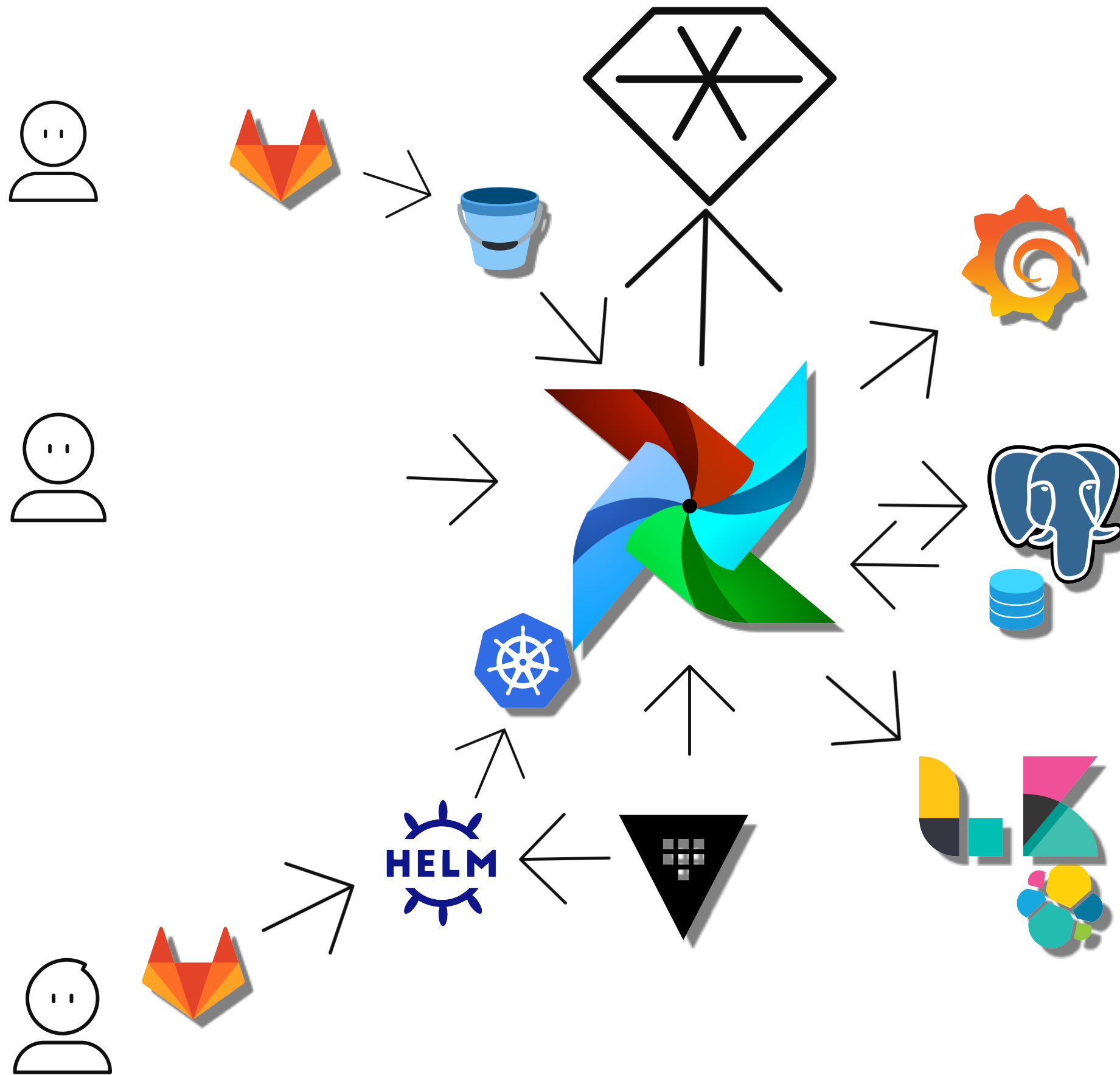


## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продам
- + Легкое добавление секретов и переменных
- + Отдельно вынесенный кластер БД с метаданными(postgres)
- + Подключен log backend(elastic)
- + Выводятся метрики + алертинг (statsd+GrafanaOnCal)

## Минусы

- Увеличение времени между созданием дага и запуском в прод
- Секреты хранятся в надежном месте



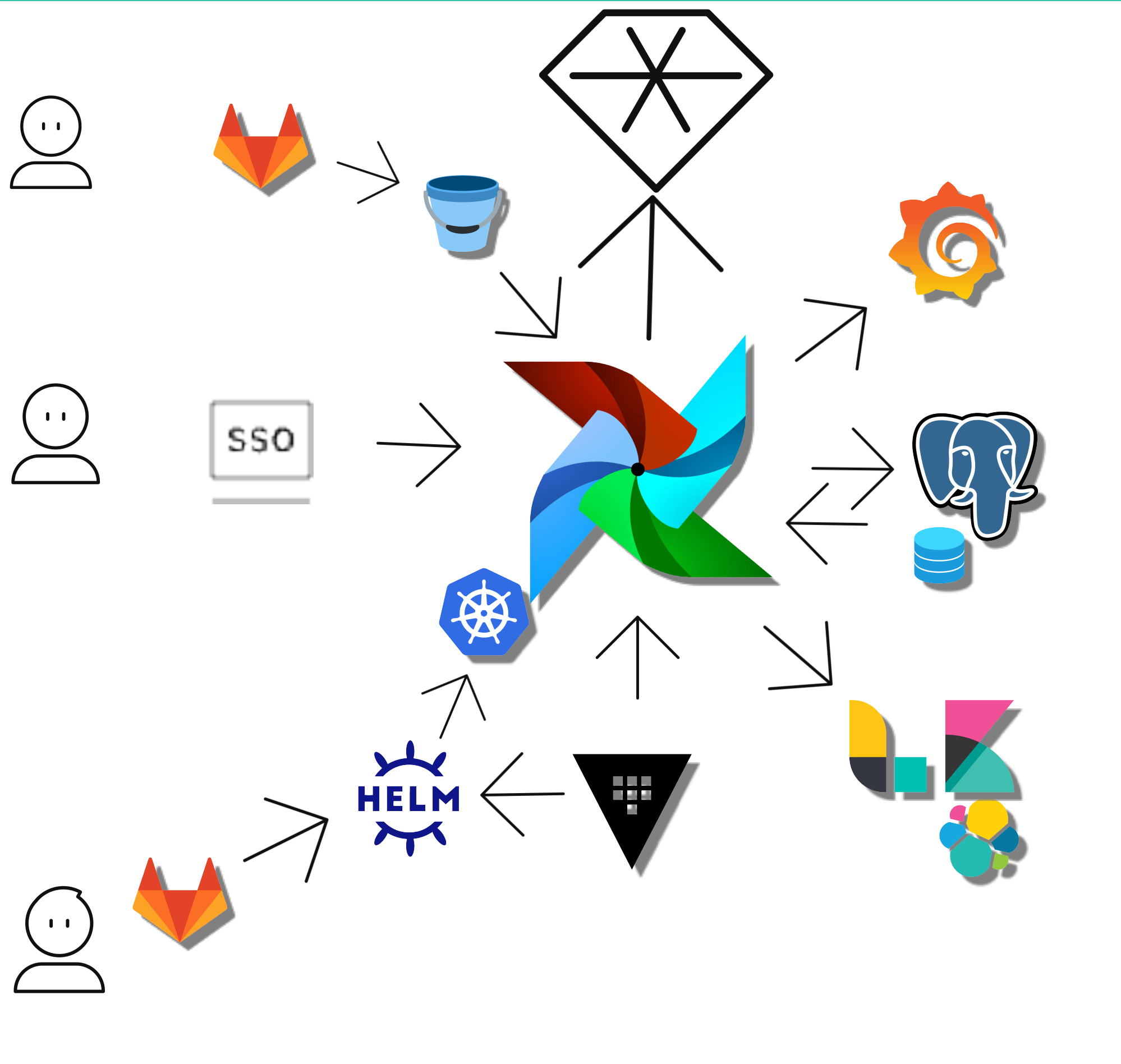
## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продам
- + Подключен secret backend для connections и секретов конфигурации(Hashicorp Vault).
- + Отдельно вынесенный кластер БД с метаданными(postgres)
- + Подключен log backend(elastic)
- + Выводятся метрики + алертинг (statsd+GrafanaOnCal)

## Минусы

- Сложное добавление коннекшенов\переменных.
- Увеличение времени между созданием дага и запуском в прод





## Плюсы

- + Airflow запускается предсказуемо и масштабируемо(kubernetes+helm), в стабильном окружении(gitlab).
- + Есть ревью дагов и тестовые среды перед продам
- + Подключен secret backend для connections и секретов конфигурации(Hashicorp Vault).
- + Отдельно вынесенный кластер БД с метаданными(postgres)
- + Вход по SSO(ADFS)
- + Подключен log backend(elastic)
- + Выводятся метрики + алертинг (statsd+GrafanaOnCal)

## Минусы

- Сложное добавление коннекшенов\переменных.
- Увеличение времени между созданием дага и запуском в прод

# ВЫВОДЫ

спасибо за внимание