

Программное обеспечение

Система управления данными Polyflow

РУКОВОДСТВО АДМИНИСТРАТОРА

Инв. № подп.	Подпись и дата	Взам. инв. №	Инв. № подубл.	Подпись и дата

Аннотация

Настоящий документ является руководством администратора системы управления данными Polyflow.

Документ разработан в соответствии с требованиями ГОСТ Р 59795-2021 «Требования к содержанию документов».

Инв. № подп.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.				
Пров.				
Н. контр.				
Утв.				

Polyflow
Руководство администратора

Лист.	Лист	Листов
	2	64
Наименование исполнителя		

.РЭ

Содержание

1 Назначение и условия применения.....	8
1.1 Назначение системы.....	8
1.2 Условия применения	8
1.2.1 Серверная часть.....	8
1.2.2 Локальная сеть.....	9
2 Подготовка к работе.....	10
2.1 Состав программного обеспечения	10
2.1.1 Установка и настройка Docker.....	10
2.1.2 Установка компонентов на один сервер	11
2.1.3 Развёртывание Polyflow в облачных средах	12
2.1.4 Гибридные конфигурации Polyflow	15
2.1.5 Кластерные конфигурации и высокая доступность.....	17
2.1.6 Описание утилиты управления сервисом Polyflow manage.py	24
2.1.7 Описание файлов и структуры проекта	28
2.1.8 Поддержка локализации.....	29
2.1.9 Настройки в airflow.cfg	29
2.1.10 Требования к СУБД хранилища	29
2.2 Порядок проверки работоспособности	31
2.3 Порядок обновления системы.....	32
3 Описание операций администрирования.....	33
3.1 Определения и сокращения Polyflow	33
3.2 Настройки компонентов платформы.....	34
3.2.1 Общие настройки компонентов платформы	34
3.2.2 Ограничение ресурсов для контейнера	34
3.2.3 Настройка схем авторизации	36
3.3 Настройка уведомлений по SMTP	41
3.3.1 Конфигурация SMTP сервера	41
3.3.2 EmailHook - системный хук для работы с email.....	42
3.3.3 Настройка шаблонов писем	43
3.3.4 Пример пользовательского шаблона HTML	43
3.3.5 Практические рекомендации	44
3.4 Операции группы безопасность.....	44
3.4.1 Список пользователей	44

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата
Изм	Лист	№ докум.	Подп.	Дата

3.4.2 Список ролей	46
3.4.3 Статистика пользователя.....	48
3.4.4 Основные разрешения	48
3.4.5 Отображения/меню	48
3.4.6 Право на просмотр/меню.....	49
3.5 Операции группы Admin	50
3.5.1 Variables	50
3.5.2 Configuration	51
3.5.3 Connections.....	51
3.5.4 Plugins.....	52
3.5.5 Providers	53
3.5.6 Pools	54
3.5.7 XComs.....	58
3.5.8 DAGs Code Editor	58
3.6 Политики безопасности и шифрования	59
3.6.1 Шифрование передаваемых данных	59
3.6.2 Рекомендации по настройке.....	60
4 Аварийные ситуации	61
5 Рекомендации по освоению	62

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Введение

Модуль Polyflow представляет собой сервис оркестровки, сбора и обработки разнородных данных в хранилищах произвольной архитектуры.

Уровень подготовки персонала, необходимого для работы с ИАС, предполагает наличие следующих групп пользователей:

- служба эксплуатации ИАС;
- разработчики;
- операторы.

Служба эксплуатации ИАС

В службу эксплуатации ИАС входят специалисты следующих категорий: «Администратор защиты (безопасности) информации», «Администратор операционных систем», «Администратор баз данных».

- Администратор защиты (безопасности) информации обеспечивает:
 - Формирование списка пользователей, допущенных к работе с Системой;
 - Настройку учетных записей пользователей, управление ролями доступа, а также интеграция пользователей с помощью LDAP;
 - Формирование матрицы доступа к ресурсам Системы и данным, а также изменение прав доступа.
- Администратор операционных систем отвечает за:
 - Установку компонентов платформы, активацию и первоначальную настройку.
 - Сопровождение ИАС (тестирование работоспособности, восстановление и т.п.), обновление версий (анализ необходимости перехода на новые версии, разработку перечня мероприятий по переводу на новую версию).
- Администратор баз данных отвечает за:

Инв. № подл.	Подпись и дата	Инв. № подл.	Взам. инв. №	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- Генерацию систем управления базами данных;
- Сопровождение и управление информационными ресурсами;
- Сохранение резервных копий, восстановление искаженной информации, архивирование информации и организацию поступления информации из архива;
- Обработку и анализ статистической информации о характере и интенсивности использования данных, о распределении нагрузки на различные компоненты структуры баз данных, внесение изменений в структуру баз данных в процессе эксплуатации Системы с целью повышения производительности, обеспечивает ввод и поддержание в актуальном состоянии общих разделов баз данных (классификаторов).

Служба эксплуатации обеспечивает функционирование в штатном режиме технических и программных средств АИС, отслеживает процессы наполнения АИС данными.

Поддержка функционирования Системы должна осуществляться силами действующей Службы эксплуатации АИС, состоящей из специалистов, обладающих знаниями в области информационных и сетевых платформ, на которых реализована АИС, и опытом администрирования баз данных.

Разработчики

Разработчиками являются специалисты, которые участвуют в процессах разработки процессов загрузки данных в Систему, описания сущностей-получателей и источников, проведения анализа работы процессов, выявления аномалий и их причин.

Разработчики должны иметь опыт разработки в своей отрасли, обладать навыками работы с DWH решениями и базовыми знаниями SQL и Python.

Инв. № подл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Операторы

Операторами являются специалисты, которые участвуют в процессах сбора данных, базовой настройки, запуска и мониторинга таких процессов.

Для работы с системой необходимо ознакомиться со следующим набором эксплуатационной документации:

- Руководство администратора;
- Руководство разработчика;
- Руководство пользователя.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

.РЭ

Лист

7

1 Назначение и условия применения

1.1 Назначение системы

Модуль Polyflow предназначен для решения следующих задач:

- сбор данных из разных источников:
 - файлы
 - внешние системы
 - базы данных
- описание сущностей-получателей и источников;
- мониторинг и управление выполняемыми процессами;
- контроль качества данных;
- трансформация данных.

1.2 Условия применения

Для функционирования АИС необходимо следующее программно-аппаратное обеспечение:

1.2.1 Серверная часть

Минимальные требования к серверному оборудованию следующие:

- CPU 2 vCPU (2.8 ГГц и выше);
- RAM 8 ГБ,
- HDD 10 ГБ

Ориентировочная формула для подсчета конфигурации в зависимости от количества процессов при использовании Local Executor: дополнительно к минимальным системным требованиям необходимо RAM 256-512МБ CPU 0.1 vCPU в среднем на каждый процесс.

Операционная система: Astra Linux Special Edition 1.6 (Смоленск) или аналог.

Права пользователя, разворачивающего приложение: user - non-root with sudo privileges.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Дополнительные требования к установленным приложениям: Docker версии 20.10.0 и до 25, Docker -compose версия 1.29 и выше, python поддерживаемой версии (<https://devguide.python.org/versions>).

1.2.2 Локальная сеть

Все компоненты платформы должны находиться в одной подсети или должна обеспечиваться прозрачная маршрутизация. Не рекомендуется использовать NAT. В рамках ознакомления рекомендуется отключить брандмауэры. Внутри локальной сети между всеми компонентами не должно быть ограничений по передаче данных. Для доступа из внешней сети достаточно открыть порт, используемый Polyflow (порт задается при установке). При использовании системы с установленными антивирусами или комплексными системами защиты необходимо обеспечить свободную работу, сетевую активность и взаимодействие компонентов.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата
--------------	----------------	--------------	--------------	----------------

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

.РЭ

Лист
9

2 Подготовка к работе

2.1 Состав программного обеспечения

Модуль Polyflow поставляется в виде нескольких файлов:

- образы Docker, содержащие в себе все компоненты с уже настроенным окружением и всеми внутренними зависимостями;
- python-файл manage.py для быстрого развертывания и настройки модуля;
- дополнительные файлы, используемые при развертывании: __init__.py, docker-compose.yml.tpl, airflow.cfg.tpl.

Примечание: Docker — программное обеспечение для автоматизации развертывания и управления приложениями в среде виртуализации на уровне операционной системы. Суть и одно из предназначений Docker такое же, как и у виртуальных машин — это изоляция работы различных конфликтующих программ внутри одного сервера. Наглядно увидеть отличия между виртуальной машиной и контейнером можно, пройдя по ссылке: <https://www.docker.com/what-docker>.

Таким образом, установка сводится к двум шагам:

1. Установка docker-engine внутри операционной системы;
2. Запуск python-файла manage.py для разворачивания готовых контейнеров с компонентами системы.

Настройка и разработка процессов загрузки и обработки данных описаны в документе «Руководство разработчика». Основные операции с интерфейсом описаны в документе «Руководство пользователя».

2.1.1 Установка и настройка Docker

1. Установить Docker в соответствии с инструкцией:

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>.

Примечание - Рекомендуемая версия 20.10.12.

2. Установить Docker Compose в соответствии с инструкцией:

<https://docs.docker.com/compose/install/>.

Инв. № подп.	Подпись	Инв. № даты	Взам. инв. №	Инв. № дубл.	Подпись и дата

Примечание - Рекомендуемая версия 1.29.2.

2.1.2 Установка компонентов на один сервер

2.1.2.1 Настройка SSL/TLS шифрования

Polyflow поддерживает сквозное шифрование данных через SSL/TLS сертификаты:

Требования к сертификатам:

- Файл сертификата: `cert.pem`
- Файл приватного ключа: `key.pem`
- Поддерживаемые алгоритмы: RSA, ECDSA

Порядок настройки:

1. Разместите файлы сертификатов в директории сервиса
2. Укажите пути к файлам при развертывании через manage.py
3. Система автоматически настроит HTTPS-доступ к веб-интерфейсу

Проверка работы:

после установки доступ к веб-интерфейсу осуществляется по HTTPS, что гарантирует шифрование всего трафика между клиентом и сервером.

2.1.2.2 Порядок разворачивания

1. Создать директорию для файлов образов

```
cd ~ && mkdir install
```

2. Скопировать из полученного дистрибутива в созданную директорию архивы базовых образов

- df_operator .tar
- df_airflow .tar
- df_postgres13 .tar

3. выполнить команды:

```
docker load -i ~/install/df_operator.tar
docker load -i ~/install/df_airflow.tar
```

Инв. № подл.	Подпись и дата
Изм	Лист

Изм	Лист	№ докум.	Подп.	Дата

```
docker load -i ~/install/df_postgres13.tar
```

4. Создать директории сервиса

```
cd ~ && mkdir dataflow && cd dataflow
```

5. Скопировать из полученного дистрибутива в созданную директорию инициализирующие компоненты

- manage.py
- __init__.py
- docker-compose.yml.tpl
- airflow.cfg.tpl

6. Разместить в созданной директории файлы ключей (key.pem) и сертификатов (cert.pem) (путь до файлов может быть задан при установке).

7. Развернуть сервис

```
python3 manage.py -deploy
```

2.1.3 Развёртывание Polyflow в облачных средах

Архитектура Polyflow, основанная на Docker-контейнерах, обеспечивает высокую переносимость и позволяет развернуть систему на любой облачной инфраструктуре, поддерживающей Docker и Docker Compose. Ниже приведены рекомендуемые конфигурации для популярных российских облачных платформ.

2.1.3.1 Конфигурация на Яндекс Облаке

Яндекс Облако предоставляет полный спектр услуг IaaS (инфраструктура как услуга) для запуска контейнеризированных приложений.

Таблица 1. Рекомендуемая конфигурация сервисов
Рекомендация для Polyflow

Компонент Яндекс Облако		Рекомендация для Polyflow	
Виртуальная машина (Compute Cloud)		Конфигурация: 2+ vCPU, 8+ ГБ RAM, 50+ ГБ SSD	

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

	Образ: Container Optimized Image или стандартный образ Ubuntu/CentOS с предустановленным Docker.
Управляемый Kubernetes	Кластер: 3 ноды (виртуальные машины) с конфигурацией не ниже 2 vCPU, 8 ГБ RAM. Преимущество: сервис управляет мастер-нодами, обеспечивая отказоустойчивость и упрощая эксплуатацию.
Docker-образы (Container Registry)	Использование: хранение и управление образами df_operator, df_airflow, df_postgres13 и другими кастомными образами.

Порядок развертывания:

- 1) Создайте виртуальную машину или кластер Managed Service for Kubernetes в Яндекс Облаке.
- 2) Установите Docker и Docker Compose на ВМ (если используется Compute Cloud).
- 3) Скопируйте файлы дистрибутива Polyflow (manage.py, docker-compose.yml.tpl и др.) на сервер.
- 4) Запустите развертывание с помощью утилиты manage.py.

2.1.3.2 Конфигурация на Selectel

Selectel предлагает надежную облачную инфраструктуру с возможностью развертывания виртуальных машин и управления контейнерами.

Таблица 2. Рекомендуемая конфигурация сервисов

Компонент Selectel	Рекомендация для Polyflow
Облачные серверы	Конфигурация: 2+ vCPU, 8+ ГБ RAM, 50+ ГБ SSD Образ: Выберите образ Ubuntu 20.04 LTS или CentOS 7 с последующей установкой Docker.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Управляемый Kubernetes	Кластер: 3 рабочие ноды с аналогичными характеристиками. Selectel полностью управляет жизненным циклом кластера.
------------------------	--

Порядок развертывания:

- 1) Активируйте облачные серверы или управляемый Kubernetes в панели управления Selectel.
- 2) Убедитесь, что на серверах установлены Docker и Docker Compose.
- 3) Разместите файлы дистрибутива Polyflow в домашней директории пользователя.
- 4) Выполните стандартную процедуру установки, как описано в разделе «Подготовка к работе».

2.1.3.3 Ключевые преимущества облачного размещения

- Масштабируемость: легко увеличивайте вычислительные ресурсы (vCPU, RAM) или добавляйте ноды в кластер Kubernetes для обработки растущей нагрузки.
- Отказоустойчивость: облачные провайдеры гарантируют высокую доступность дата-центров и сервисов. При использовании Kubernetes достигается отказоустойчивость на уровне приложения.
- Управляемость: использование управляемых сервисов (Kubernetes, реестр контейнеров) снижает операционную нагрузку на администраторов.
- Экономическая эффективность: оплата происходит только за потребленные ресурсы, что позволяет оптимизировать затраты.

2.1.3.4 Общие рекомендации

- Безопасность: настройте Security Groups (группы безопасности) для строгого ограничения входящего и исходящего трафика. Откройте для доступа только порт, используемый веб-интерфейсом Polyflow (по умолчанию 8080).

Инв. № подл.	Подпись	Инв. № даты	Взам. инв. №	Инв. № дубл.	Подпись и дата
--------------	---------	-------------	--------------	--------------	----------------

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

.РЭ

Лист

- Резервное копирование: регулярно используйте встроенную утилиту manage.py --backup для создания резервных копий. Также настройте сnapshotы дисков виртуальных машин средствами облачного провайдера.

Сетевые настройки: убедитесь, что все компоненты Polyflow (веб-сервер, планировщик, исполнители) находятся в одной подсети или обеспечена прозрачная маршрутизация между ними, как того требуют условия применения.

2.1.4 Гибридные конфигурации Polyflow

Архитектура Polyflow, основанная на изолированных Docker-контейнерах, органично поддерживает гибридные сценарии развертывания, позволяя распределять компоненты системы между различными средами (например, частным облаком и публичным облаком или локальной инфраструктурой) в зависимости от требований к безопасности, производительности и стоимости.

2.1.4.1 Пример гибридной конфигурации: "Частное облако + Публичное облако"

Данная модель идеально подходит для задач, где критически важные данные и основные вычислительные нагрузки должны оставаться под полным контролем внутри частного облака или локального ЦОД, при этом требуется эластичное масштабирование для обработки пиковых нагрузок.

Сценарий: основные компоненты Polyflow размещены в частном облаке заказчика, а дополнительные воркеры (Airflow Workers) для выполнения задач ETL запускаются в публичном облаке (например, Яндекс Облако) для обработки всплесков нагрузки.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Таблица 3. Пример гибридной конфигурации

Компонент	Рекомендуемая среда размещения	Обоснование
Веб-сервер (Webserver)	Частное облако / Локальный ЦОД	Обеспечение безопасного доступа к UI, соблюдение требований к внутреннему шлюзу.
Планировщик (Scheduler)	Частное облако / Локальный ЦОД	Централизованное управление выполнением DAG, требует минимальной задержки с БД метаданных.
База данных метаданных (PostgreSQL)	Частное облако / Локальный ЦОД	Хранение критических метаданных и информации о выполнении задач в защищенной среде.
Воркеры (Workers)	Публичное облако (Яндекс Облако, Selectel)	Эластичное масштабирование: быстрое развертывание дополнительных воркеров для параллельного выполнения большого числа задач.

Ключевые технические требования:

- Сетевое соединение: необходимо обеспечить защищенное сетевое соединение (например, VPN-туннель или выделенный канал связи) между сегментами частного и публичного облака.
- Доступ к данным: Воркеры в публичном облаке должны иметь сетевой доступ к источникам данных и получателям, а также к очереди сообщений (например, Redis) и БД метаданных в частном облаке.

Преимущества гибридного подхода для Polyflow

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- Балансировка нагрузки и масштабируемость: вы можете поддерживать базовый пул воркеров в частной инфраструктуре и динамически добавлять "вычислительные узлы" в облаке в периоды пиковой нагрузки, например, при ночных запусках сложных ETL-процессов.
- Оптимизация затрат: гибридная модель позволяет избежать капитальных затрат на мощное "железо", которое будет простоявать большую часть времени. Вы оплачиваете облачные ресурсы только на время их фактического использования.
- Безопасность и соответствие требованиям: критические компоненты (БД метаданных, планировщик) остаются внутри контролируемого периметра безопасности компании, что помогает соблюдать строгие нормативы, такие как 152-ФЗ.

Рекомендации по реализации

- 1) Единая точка управления: Для упрощения развертывания и управления воркерами в облаке рекомендуется использовать единые шаблоны Docker Compose и систему управления конфигурациями.
- 2) Совместимость образов: Убедитесь, что Docker-образ `df_operator`, используемый для выполнения задач, идентичен в обеих средах.
- 3) Резервное копирование: Процедуры бэкапа, выполняемые утилитой `manage.py --backup`, должны оставаться нацеленными на БД метаданных в частном облаке.

2.1.5 Кластерные конфигурации и высокая доступность

Архитектура Polyflow поддерживает развертывание в кластерных конфигурациях, что обеспечивает отказоустойчивость, горизонтальное масштабирование и высокую доступность системы.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

2.1.5.1 Архитектура высокодоступного кластера

Polyflow поддерживает следующие режимы кластеризации компонентов:

Таблица 4. Режимы кластеризации компонентов

Компонент	Режим кластеризации	Конфигурация
Веб-сервер	Активный-активный	Несколько экземпляров за балансировщиком нагрузки
Планировщик	Активный-резервный	Один активный планировщик, несколько standby
Воркеры	Активный-активный	Неограниченное количество воркеров в пуле
База метаданных	Кластерная СУБД	PostgreSQL с репликацией или MS SQL Server Always On.

2.1.5.2 Конфигурация кластера с CeleryExecutor

Рекомендуемая конфигурация для продакшн-среды:

```
#yaml
# docker-compose-ha.yml
version: '3'
services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_DB: airflow
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow

  redis:
    image: redis:6-alpine
    command: redis-server --appendonly yes
```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

```

webserver:
  image: df_airflow:latest
  deploy:
    replicas: 2
    command: webserver
    environment:
      AIRFLOW__CORE__EXECUTOR: CeleryExecutor
      AIRFLOW__DATABASE__SQLALCHEMY_CONN:
        postgresql+psycopg2://airflow:airflow@postgres/airflow
      AIRFLOW__CELERY__BROKER_URL: redis://redis:6379/0
  depends_on:
    - postgres
    - redis

scheduler:
  image: df_airflow:latest
  deploy:
    replicas: 1
    command: scheduler
    environment:
      AIRFLOW__CORE__EXECUTOR: CeleryExecutor
      AIRFLOW__DATABASE__SQLALCHEMY_CONN:
        postgresql+psycopg2://airflow:airflow@postgres/airflow
      AIRFLOW__CELERY__BROKER_URL: redis://redis:6379/0
  depends_on:
    - postgres
    - redis

worker:
  image: df_operator:latest
  deploy:
    replicas: 4
    command: worker
    environment:
      AIRFLOW__CORE__EXECUTOR: CeleryExecutor
      AIRFLOW__DATABASE__SQLALCHEMY_CONN:
        postgresql+psycopg2://airflow:airflow@postgres/airflow

```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

```

AIRFLOW__CELERY__BROKER_URL: redis://redis:6379/0

depends_on:
  - postgres
  - redis

```

2.1.5.3 Кластерная конфигурация в Kubernetes

Пример манифеста для развертывания в Kubernetes:

```

#yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: airflow-webserver
spec:
  replicas: 2
  selector:
    matchLabels:
      app: airflow-webserver
  template:
    metadata:
      labels:
        app: airflow-webserver
    spec:
      containers:
        - name: webserver
          image: df_airflow:latest
          command: ["webserver"]
      env:
        - name: AIRFLOW__CORE__EXECUTOR
          value: "KubernetesExecutor"
        - name: AIRFLOW__DATABASE__SQLALCHEMY_CONN
          valueFrom:
            secretKeyRef:
              name: airflow-secrets
              key: database_url
  ---
apiVersion: apps/v1

```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

```

kind: Deployment

metadata:
  name: airflow-scheduler

spec:
  replicas: 2
  selector:
    matchLabels:
      app: airflow-scheduler
  template:
    metadata:
      labels:
        app: airflow-scheduler
    spec:
      containers:
        - name: scheduler
          image: df_airflow:latest
          command: ["scheduler"]
      env:
        - name: AIRFLOW__CORE__EXECUTOR
          value: "KubernetesExecutor"

```

2.1.5.4 Преимущества кластерной конфигурации

Отказоустойчивость:

- Автоматическое переключение при отказе компонентов
- Резервирование критических сервисов
- Сохранение состояния выполнения задач

Масштабируемость:

- Динамическое добавление воркеров под нагрузку
- Распределение задач между узлами кластера
- Балансировка нагрузки на веб-интерфейс

Обслуживание:

- Бесперебойное обновление компонентов
- "Горячее" добавление мощностей

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- Минимизация downtime

2.1.5.5 Рекомендации по настройке

- 1) База данных: Используйте кластерную СУБД с репликацией
- 2) Брокер сообщений: Redis Sentinel или RabbitMQ Cluster для высокой доступности
- 3) Сеть: Обеспечьте низкую задержку между узлами кластера
- 4) Мониторинг: Настройте health checks для всех компонентов
- 5) Резервное копирование: Регулярные бэкапы БД метаданных

2.1.5.6 Требования к инфраструктуре

- Минимальная конфигурация: 3 узла для отказоустойчивости
- Сетевая связность: < 1ms задержка между узлами
- Дисковое пространство: Раздельные тома для данных и логов
- Безопасность: TLS-шифрование трафика между компонентами

2.1.5.7 Принципы отказоустойчивости компонентов

Обеспечение отказоустойчивости (Failover)

Архитектура Polyflow позволяет настроить отказоустойчивую конфигурацию, которая обеспечивает непрерывность работы сервиса даже в случае сбоя отдельных компонентов. Это достигается за счет дублирования критических служб и использования кластерных решений.

Ключевые компоненты и принципы их резервирования:

1) Веб-сервер (Webserver)

- Роль: предоставляет пользовательский интерфейс для управления и мониторинга рабочих процессов (DAGs).
- Обеспечение отказоустойчивости: может быть развернут в нескольких экземплярах в конфигурации «активный-активный» (active-active). Для распределения нагрузки и единой точки входа между экземплярами используется балансировщик нагрузки.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

2) Планировщик (Scheduler)

- Роль: отслеживает расписания и запускает задачи.
- Обеспечение отказоустойчивости: хотя планировщик является ключевым компонентом, для него может быть настроена схема «активный-резервный» (active-passive). В этом случае работает один основной планировщик, в то время как резервный находится в режиме ожидания и готов взять на себя функции в случае сбоя основного. Это позволяет избежать единой точки отказа.

3) Исполнители (Workers/Executors)

- Роль: непосредственно выполняют задачи рабочих процессов.
- Обеспечение отказоустойчивости: развертываются в конфигурации «активный-активный». Количество воркеров может масштабироваться горизонтально. При использовании CeleryExecutor задачи автоматически перераспределяются между живыми узлами кластера в случае падения одного из них.

4) База данных метаданных (Metadata Database)

- Роль: хранит состояние всех DAG, задач, переменных и подключений.
- Обеспечение отказоустойчивости: для базы данных (например, PostgreSQL) необходимо настроить кластерную конфигурацию с репликацией. Это гарантирует, что в случае отказа основной базы данных одна из реплик сможет принять на себя нагрузку, обеспечивая сохранность и доступность метаданных.

5) Брокер сообщений (Message Broker)

- Роль: обеспечивает очередь задач для CeleryExecutor.

Инв. № подл.	Подпись	Инв. № даты	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата
--------------	---------	-------------	----------------	--------------	--------------	----------------

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

- Обеспечение отказоустойчивости: брокеры, такие как Redis или RabbitMQ, должны быть развернуты в кластерном режиме высокой доступности (HA). Для RabbitMQ это включает создание кластера зеркалирования очередей, чтобы сообщения не были потеряны при сбое узла.

Общая архитектура отказоустойчивого кластера:

- В высокой доступности компоненты объединяются в единую систему:
- Веб-серверы размещаются за балансировщиком нагрузки.
- Планировщики работают в режиме активный-резервный.
- Воркеры объединяются в пул под управлением CeleryExecutor.
- База данных и брокер работают в своих отказоустойчивых кластерных конфигурациях.

Настройка такого кластера в среде Kubernetes дополнительно упрощается за счет использования официального Helm-чарта Airflow, который помогает развернуть и сконфигурировать все необходимые компоненты.

2.1.6 Описание утилиты управления сервисом Polyflow manage.py

Автоматизирует процесс установки, обновления и управления Polyflow.

Принимает следующие аргументы:

- `'-i` или `--interactive` интерактивный режим обновления
- `--reconfigure` сгенерировать файлы конфигураций
- `--deploy` развернуть сервисы с нуля
- `--update` обновить сервисы
- `--compile` подготовка структуры артефактов

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- `--run-dags` запуск системных dag'ов `df_datadb_upgrade` и `df_datadb_populate` (для запуска необходимо наличие системного подключения)
- `--sync-perm` синхронизации разрешений и добавление ролей
- `--df-conn-id` идентификатор системного подключения, по умолчанию: `df`
- `--backup` бэкап бд и конфигурации
- `--restore` восстановление бд и конфигурацию
- `--backup-ws` бэкап рабочего пространства, содержащего системный исходный код и артефакты проектов
- `--restore-ws` восстановление рабочего пространства, содержащего системный исходный код и артефакты проектов
- `--init-project` создание структуры проекта
- `--no-rebuild` не пересобирать локальные образы
- `--no-restart` не запускать (перезапускать) сервисы
- `--no-upgrade` не обновлять базу данных
- `--docker-repo` задать репозиторий docker-образов, например: `test.polymedia.ru/r5/`
- `--timeout` задать таймаут выполнения шагов утилиты в секундах (по умолчанию 10 секунд)
- `--localize` пересобрать библиотеки интерфейса (в том числе появляется возможность скомпилировать пользовательские файлы переводов)
- `--rm-default-conns` удалить примеры подключений Airflow
- `--rm-temp-tables` удалить временные таблицы Airflow.

Примеры запуска:

- `python3 manage.py --backup --no-restart --no-rebuild --no-upgrade` создать бэкап базы и конфигурации сервиса (при обновлении)

Инв. № подл.	Подпись	Инв. № даты	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- `python3 manage.py --update --backup --no-upgrade --compile` запустить обновление
- `python3 manage.py --deploy` развернуть сервис
- `python3 manage.py --run-dags --no-restart --no-rebuild --no-upgrade` запустить системные dag'и
- `python3 manage.py --no-restart --no-upgrade` пересобрать образы
- `python3 manage.py --no-restart --no-upgrade --localize` пересобрать образы с компиляцией библиотек интерфейса
- `python3 manage.py --add-df-conn --no-restart --no-rebuild --no-upgrade` создать подключение `df` к базе данных со схемой с системными объектами.

2.1.6.1 Резервное копирование

Резервное копирование предусматривает использование утилиты `manage.py`. Сервис при этом останавливается.

Доступно создание резервных копий базы и конфигурации сервиса, а также рабочего пространства.

При создании и восстановлении из резервных копий по умолчанию файлы сохраняются в папке `backups` директории сервиса.

2.1.6.1.1 Создание бэкапа базы и конфигурации сервиса

Пример создания бэкапа базы и конфигурации сервиса в указанную директорию с последующим запуском сервиса:

- `python3 manage.py --backup ~/backups/polyflow/ --no-rebuild --no-upgrade`

При этом будут созданы:

- копии файлов:
 - `Dockerfile_operator`
 - `airflow.env`

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- docker-compose.yml
- airflow.cfg
- дамп базы airflow с помощью инструмента pg_dump.

Пример восстановления из директории с резервными копиями с последующим запуском сервиса:

- `python3 manage.py --restore ~/backups/polyflow/ --no-rebuild --no-upgrade`

Восстановление происходит из последней резервной копии.

2.1.6.1.2 Создание бэкапа рабочего пространства

Пример создания бэкапа рабочего пространства, содержащего системный исходный код и артефакты проектов в указанную директорию с последующим запуском сервиса:

- `python3 manage.py --backup-ws ~/backups/polyflow/ --no-rebuild --no-upgrade`

При этом будут созданы архивы папок `dags`, `metadata`, `plugins`, `source` тома `df_workspace` (по умолчанию в папке `volumes/workspace` из директории сервиса).

Пример восстановления из директории с резервными копиями с последующим запуском сервиса:

- `python3 manage.py --restore-ws ~/backups/polyflow/ --no-rebuild --no-upgrade`

Восстановление происходит из последней резервной копии.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

В некоторых случаях (например, при ошибке 'Operation not permitted') команды должны быть запущены с использованием привилегий суперпользователя: 'sudo python3 manage.py --backup-ws' или 'sudo python3 manage.py --restore-ws'. Это позволит корректно выполнить процесс резервного копирования и восстановления.

2.1.7 Описание файлов и структуры проекта

Вся работа происходит в папке projects, содержимое которого линкуется с папкой workspace.

Для инициализации структуры проекта необходимо выполнить команду `python3 manage.py --init-project <Имя проекта>`. Если имя проекта не задано, то будет использовано значение default. После выполнения команды будут созданы следующие папки:

- dataflow/volumes/projects/<Имя проекта>/cache
- dataflow/volumes/projects/<Имя проекта>/dags
- dataflow/volumes/projects/<Имя проекта>/metadata
- dataflow/volumes/projects/<Имя проекта>/plugins
- dataflow/volumes/projects/<Имя проекта>/share
- dataflow/volumes/projects/<Имя проекта>/source

Дополнительно после установки сервиса доступны следующие конфигурационные файлы:

- airflow.env - переменные окружения, используемые airflow;
- operator.env - переменные окружения, используемые в operator (например, схемы БД);
- airflow.cfg - конфигурация планировщика и веб-интерфейса.

Тома:

- /home/osuser/docker/dataflow/volumes/projects - том df_projects;

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

- /home/osuser/docker/dataflow/volumes/workspace - том df_workspace;
- /home/osuser/docker/dataflow/volumes/data - том df_data;
- /home/osuser/docker/dataflow/volumes/logs - том df_logs.

Прикладная разработка и оператор работают в df_projects, airflow работает с df_workspace.

2.1.8 Поддержка локализации

В приложении реализована интернационализация и поддержан русский язык. Для переключения языка в airflow.cfg для настройки babel_default_locale необходимо указать значение ru.

Поддерживаются пользовательские словари перевода. Для их подключения в локальном Dockerfile airflow необходимо скопировать папку с переводами в соответствующую папку сервиса. Пример команды: `COPY --chown=airflow:airflow translations home/airflow/.local/lib/python3.9/site-packages/airflow/www/translations.`

Папка с переводами должна иметь следующую структуру translations/<locale>/LC_MESSAGES/messages.po, где:

- <locale> - код локали, например: ru, zh, en;
- messages.po - файл перевода, который необходимо создать на основе шаблона (можно получить командой `docker cp df-webserver:/home/airflow/.local/lib/python3.9/site-packages/airflow/www/translations/messages.pot ./messages.pot`).

2.1.9 Настройки в airflow.cfg

При разворачивании приложения на тестовых серверах допустимо использовать http. В этом случае настройку cookie_secure можно установить в False.

2.1.10 Требования к СУБД хранилища

Поддерживается СУБД PostgreSQL 12+.

Инв. № подл.	Подпись и дата
Взам. инв. №	Подпись и дата
Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Для корректной работы модулей системные таблицы рекомендуется создавать в отдельной схеме dwh, установленной по умолчанию для пользователя dwh.

Создание и заполнение системных таблиц выполняется последовательным запуском DAG'ов df_upgrade_datadb и df_populate_datadb.

Для работы необходимо добавить объект Connection:

```
Conn Id: df
Conn Type: <Необходимый тип базы данных хранилища>
Host: <Хост базы данных>
Login: <Логин>
Password: <Пароль>
Extra: {"schema": "<Схема приложения>"}
```

Подключение к базе данных с системными таблицами автоматически доступно в DockerOperator через переменную окружения `AF_DF_CONNECTION`.

Переопределить наименование схемы df возможно через переменную `AF_DF_SCHEMA`.

Если пользователь базы хранилища отличается от пользователя df, ему требуется явная выдача разрешений на системные таблицы, пример запроса:

```
grant select, insert, update, delete on all tables in schema df to dwh;
grant select, usage on all sequences in schema df to dwh;
grant execute on all functions in schema df to dwh;
grant execute on all procedures in schema df to dwh;
grant usage on schema df to dwh;
revoke select, insert, update, delete on df.alembic_version from dwh;
```

На базе должны быть доступны следующие расширения:

- в схеме `df` - расширение `dblink`:

```
```sql
create extension if not exists "dblink" schema df;
```
```

- в схеме `public` - расширение `uuid-ossp`:

```
```sql
```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

```
create extension if not exists "uuid-ossp";
```
```

Хранение данных допустимо также на СУБД MS SQL Server 2008 R2 SP3 10.50.6560.0+.

2.2 Порядок проверки работоспособности

Для проверки работы Polyflow необходимо произвести аутентификацию в системе. После успешной аутентификации откроется главная страница приложения, со списком доступных процессов (Рисунок 1). Необходимо выбрать один из них и перейти на форму просмотра информации о выбранном процессе, кликнув на его название (Рисунок 2).

Рисунок 1. Список доступных процессов

Будет открыта форма информации и управления выбранным DAG. По умолчанию открыта вкладка График (Рисунок 2).

| | | | |
|--------------|----------------|--------------|----------------|
| Инв. № подл. | Подпись и дата | Инв. № подл. | Подпись и дата |
| | | | |

| | | | | |
|-----|------|----------|-------|------|
| Изм | Лист | № докум. | Подп. | Дата |
| | | | | |

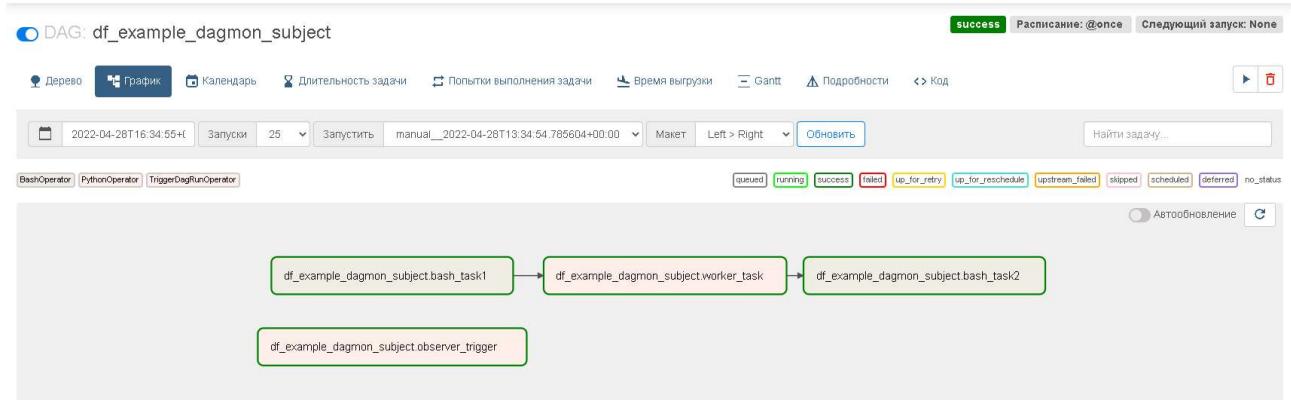


Рисунок 2. Форма просмотра информации о процессе

2.3 Порядок обновления системы

Ниже указан общий порядок обновления:

1. запустить обновление

```
python3 manage.py --update --backup --no-upgrade
```

2. перезапустить сервис

```
docker-compose down && docker-compose up -d
```

| РЭ | | | | |
|--------------|----------------|--------------|--------------|----------------|
| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № подл. | Подпись и дата |
| Изм | Лист | № докум. | Подп. | Дата |
| | | | | |

3 Описание операций администрирования

3.1 Определения и сокращения Polyflow

Определения и сокращения Polyflow представлены в таблице 1.

Таблица 5. Определения Polyflow

| Термин/Сокращение | Определение |
|--|---|
| Аутентификация | Проверка принадлежности пользователю указанного им пароля. |
| Пользователь | Авторизованный пользователь, учетная запись которого позволяет просматривать данные на портале. |
| Веб-интерфейс | Сайт в компьютерной сети, который предоставляет пользователю интерактивный интернет-сервис, который работает в рамках этого сайта. |
| Планировщик (Airflow Scheduler) | Компонент, который отслеживает состояние DAG и запускает задачи, зависимости которых были удовлетворены. После запуска системы планировщик работает непрерывно, чтобы отслеживать и синхронизировать папку, содержащую объекты DAG. |
| Хранилище данных (англ. Content Repository, Data Warehouse, DWH) | Предметно-ориентированная информационная база данных, сочетающая в себе функции системы управления версиями, поисковой машины и СУБД. |
| Система управления данными Polyflow | Сервис оркестровки сбора и обработки разнородных данных хранилища произвольной архитектуры. |
| Polyflow | Краткое наименование программного обеспечения «Система управления данными Polyflow» |
| DAG (Directed Acyclic Graph) | Смысловое объединение задач, которые необходимо выполнить в строго определенной последовательности согласно указанному расписанию. |
| Task | Операции, применяемые к данным, например: загрузка данных из различных источников, их агрегирование, индексирование, очистка от |

| | |
|--------------|----------------|
| Инв. № подл. | Подпись и дата |
| | |

| | | | | |
|-----|------|----------|-------|------|
| Изм | Лист | № докум. | Подп. | Дата |
| | | | | |

| | |
|----------|---|
| | дубликатов, сохранение полученных результатов и прочие ETL-процессы. |
| OpenID | Открытый стандарт децентрализованной системы аутентификации, предоставляющей пользователю возможность создать единую учётную запись для аутентификации на множестве не связанных друг с другом интернет-ресурсов, используя услуги третьих лиц. |
| Operator | Сущность, на основе которой создаются экземпляры заданий, где описывается, что будет происходить во время исполнения экземпляра задания. |
| Sensor | Тип оператора, позволяющий описывать реакцию на определенное событие. |
| ETL | (от англ. Extract, Transform, Load – дословно «извлечение, преобразование, загрузка») – один из основных процессов в управлении хранилищами данных. |
| SLA | (от англ. Service Level Agreement) – соглашение об уровне сервиса. |

3.2 Настройки компонентов платформы

3.2.1 Общие настройки компонентов платформы

Перед установкой необходимо проверить наличие файлов ключей и сертификатов key.pem и cert.pem, пути до которых необходимо будет указать в ходе установки.

3.2.2 Ограничение ресурсов для контейнера

Для ограничения ресурсов контейнера необходимо внести изменения в файл docker-compose.yml:

- resources: настраивает ограничения физических ресурсов для запуска контейнера на платформе. Эти ограничения могут быть настроены как:

| | |
|--------------|----------------|
| Инв. № подл. | Подпись и дата |
| | |

| | | | | |
|-----|------|----------|-------|------|
| Изм | Лист | № докум. | Подп. | Дата |
| | | | | |

- limits: платформа должна запрещать контейнеру выделять больше.
- reservations: платформа должна гарантировать, что контейнер может выделить как минимум заданный объем.

```

services:
  frontend:
    image: awesome/webapp
  deploy:
    resources:
      limits:
        cpus: '0.50'
        memory: 50M
        pids: 1
      reservations:
        cpus: '0.25'
        memory: 20M

```

- cpus настраивает ограничение или резервирование того, сколько доступных ресурсов ЦП (как количество ядер) может использовать контейнер.
- memory настраивает ограничение или резервирование объема памяти, который может выделять контейнер, в виде строки, выражающей значение байта .
- pids настраивает ограничение PID контейнера, заданное как целое число.

| | | | | |
|--------------|----------------|--------------|--------------|----------------|
| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
| | | | | |

| | | | | |
|-----|------|----------|-------|------|
| Изм | Лист | № докум. | Подп. | Дата |
| | | | | |

3.2.3 Настройка схем авторизации

3.2.3.1 Аутентификация через сервис, поддерживающий OpenID Connect

Приведен пример настройки oauth2 sso polyflow через keycloak (настройка на промышленном решении должна выполняться с учетом особенностей целевой среды):

1. развернуть keycloak (<18 или --spi-login-protocol-openid-connect-legacy-logout-redirect-uri=true, пока не вышла 19 с logout client_id);
2. создать клиент в выбранном пространстве (Realm -> Configure -> Clients -> Create):
 - o id: dataflow
 - o protocol: openid-connect
3. настроить клиент (Clients -> Dataflow -> Settings):
 - o access type: confidential
 - o authorization enabled: on
 - o valid redirect urls: https://dataflow_host:dataflow_port/*
4. создать роли (Clients -> Dataflow -> Roles -> Add Role):
 - o dataflow_admin
 - o dataflow_op
 - o dataflow_user
 - o dataflow_view
 - o dataflow_public
5. создать группы (Realm -> Manage -> Groups):
 - o dataflow_admin
 - o dataflow_op
 - o dataflow_user
 - o dataflow_view
 - o dataflow_public
6. связать роли и группы (Group -> Edit -> Role Mappings);

| | | | | |
|--------------|----------------|--------------|--------------|----------------|
| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
| Изм | Лист | № докум. | Подп. | Дата |

- для каждой созданной группы Client Roles -> dataflow -> Available Roles -> select role -> Add selected;

7. при создании пользователя включить его в соответствующую группу;
8. тестовый пример запроса токена:

```
curl -s -X POST https://keycloak.local:7070/realms/testrealm/protocol/openid-connect/token \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'client_id=dataflow' \
--data-urlencode 'client_secret=Bc4tBp05HoUXZxOTigXhY9os4sXSNxCZ' \
--data-urlencode 'username=admin' \
--data-urlencode 'password=somepass' | jq
```

9. создать файл webserver_config.py в директории dataflow. Пример webserver_config.py:

```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.

"""
webserver_config

References
```

| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
|--------------|----------------|--------------|--------------|----------------|
| | | | | |

| Изм | Лист | № докум. | Подп. | Дата |
|-----|------|----------|-------|------|
| | | | | |

```

- https://flask-
appbuilder.readthedocs.io/en/latest/security.html#authentication-oauth

"""

import os
import logging
import jwt

from flask import redirect, session, request
from flask_appbuilder import expose
from flask_appbuilder.security.views import AuthOAuthView
from airflow.www.fab_security.manager import AUTH_OAUTH, AUTH_DB
from airflow.www.security import AirflowSecurityManager

basedir = os.path.abspath(os.path.dirname(__file__)) # default
log = logging.getLogger(__name__)

DATAFLOW_HOST = 'https://dataflow.local:8080'
PROVIDER = 'keycloak'

PROVIDER_HOST = 'https://keycloak.local:7070' # visible from dataflow docker
network

REALM = 'testrealm'
CLIENT_ID = 'dataflow'
CLIENT_SECRET = 'Bc4tBp05HoUXZxOTigXhY9os4sXSNxCZ' # set in provider
KEYCLOAK_BASE_URL = f'{PROVIDER_HOST}/realms/{REALM}/protocol/openid-connect'
KEYCLOAK_TOKEN_URL = f'{KEYCLOAK_BASE_URL}/token'
KEYCLOAK_AUTH_URL = f'{KEYCLOAK_BASE_URL}/auth'
KEYCLOAK_LOGOUT_URL = f'{KEYCLOAK_BASE_URL}/logout'

WTF_CSRF_ENABLED = True # default = True
AUTH_TYPE = AUTH_OAUTH # default = AUTH_DB
AUTH_USER_REGISTRATION = False
AUTH_USER_REGISTRATION_ROLE = "Public"
AUTH_ROLES_SYNC_AT_LOGIN = True # sync roles on each login or only on
registration
PERMANENT_SESSION_LIFETIME = 43200

AUTH_ROLES_MAPPING = {

```

| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
|--------------|----------------|--------------|--------------|----------------|
| Изм | Лист | № докум. | Подп. | Дата |

```

    "dataflow_admin": ["Admin"],
    "dataflow_op": ["Op"],
    "dataflow_user": ["User"],
    "dataflow_viewer": ["Viewer"],
    "dataflow_public": ["Public"],
}

OAUTH_PROVIDERS = [
{
    'name': 'keycloak',
    'icon': 'fa-key',
    'token_key': 'access_token',
    'remote_app': {
        'client_id': CLIENT_ID,
        'client_secret': CLIENT_SECRET,
        'client_kwargs': {
            'scope': 'email profile'
        },
        'api_base_url': KEYCLOAK_BASE_URL,
        'request_token_url': None,
        'access_token_url': KEYCLOAK_TOKEN_URL,
        'authorize_url': KEYCLOAK_AUTH_URL,
    },
},
]

class CustomAuthRemoteUserView(AuthOAuthView):
    @expose("/logout/")
    def logout(self):
        """Delete access token before logging out."""
        log.info('Logout')
        super().logout()
        return
    redirect(f'{KEYCLOAK_LOGOUT_URL}?redirect_uri={DATAFLOW_HOST}')

```

| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
|--------------|----------------|--------------|--------------|----------------|
| | | | | |
| | | | | |
| | | | | |

```

class CustomSecurityManager(AirflowSecurityManager):
    authoauthview = CustomAuthRemoteUserView

    def oauth_user_info(self, provider, response=None):
        if provider == PROVIDER:
            token = response["access_token"]
            me = jwt.decode(token, algorithms="RS256", verify=False)
            groups = me["resource_access"]["dataflow"]["roles"] # unsafe

            if len(groups) < 1:
                groups = ["dataflow_public"]
            else:
                groups = [str for str in groups if "dataflow" in str]

        userinfo = {
            "username": me.get("preferred_username"),
            "email": me.get("email"),
            "first_name": me.get("given_name"),
            "last_name": me.get("family_name"),
            "role_keys": groups,
        }
        log.debug("user info: {}".format(userinfo))
        return userinfo
    else:
        return {}

SECURITY_MANAGER_CLASS = CustomSecurityManager
APP_THEME = "cerulean.css" # default "bootstrap-theme.css"

```

10. подключить файл `webserver_config.py` как том в `docker-compose.yml`:

```

volumes:
    - ./webserver_config.py:/home/airflow/webserver_config.py

```

11. в `airflow.cfg` в секции `webserver` установить `cookie_samesite = lax`;

12. перезапустить сервис `dataflow`.

| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
|--------------|----------------|--------------|--------------|----------------|
| | | | | |
| | | | | |

3.3 Настройка уведомлений по SMTP

3.3.1 Конфигурация SMTP сервера

Polyflow поддерживает отправку email-уведомлений через SMTP-сервер. Настройка производится через конфигурационный файл airflow.cfg в секции [smtp]:

```
#ini
[smtp]
# Хост SMTP сервера
smtp_host = smtp.yourcompany.com
# Порт SMTP сервера
smtp_port = 587
# Логин для аутентификации
smtp_user = your_email@yourcompany.com
# Пароль для аутентификации
smtp_password = your_password
# Использовать STARTTLS
smtp_starttls = True
# Использовать SSL
smtp_ssl = False
# Email отправителя
smtp_mail_from = polyflow@yourcompany.com
```

Эквивалентные переменные окружения:

```
#bash
AIRFLOW__SMTP__SMTP_HOST=smtp.yourcompany.com
AIRFLOW__SMTP__SMTP_PORT=587
AIRFLOW__SMTP__SMTP_USER=your_email@yourcompany.com
AIRFLOW__SMTP__SMTP_PASSWORD=your_password
AIRFLOW__SMTP__SMTP_STARTTLS=True
AIRFLOW__SMTP__SMTP_SSL=False
AIRFLOW__SMTP__SMTP_MAIL_FROM=polyflow@yourcompany.com
```

| Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата |
|--------------|----------------|--------------|--------------|----------------|
| | | | | |

| Изм | Лист | № докум. | Подп. | Дата |
|-----|------|----------|-------|------|
| | | | | |

3.3.2 EmailHook - системный хук для работы с email

Polyflow включает расширенный EmailHook, который предоставляет дополнительные возможности для работы с email-уведомлениями.

Особенности EmailHook в Polyflow

- Поддержка пользовательских шаблонов: Шаблоны располагаются относительно директории source/df/common/email
- Расширенные переменные шаблонов: Помимо стандартных переменных Airflow доступны:
 - df_now - текущая дата и время в UTC
 - base_url - значение параметра base_url из секции [webserver] конфига airflow.cfg
 - log_url - ссылка на лог в интерфейсе приложения
- Поддержка Jinja-шаблонов: Для тем и содержимого писем

Использование EmailHook в DAG

```
#python

from airflow import DAG
from datetime import datetime
from plugins.pf.hooks.email_hook import EmailHook

def send_custom_notification():
    email_hook = EmailHook()

    email_hook.send_email_smtp(
        to='recipient@company.com',
        subject='Уведомление от Polyflow',
        html_content='''<h3>Статус выполнения процесса</h3>
<p>Процесс завершен успешно</p>
<p>Время выполнения: {{ df_now }}</p>
<p><a href="{{ log_url }}">Просмотреть логи</a></p>''',
        files=['/path/to/attachment.pdf']
    )
```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

```

with DAG('email_notification_dag',
         start_date=datetime(2024, 1, 1),
         schedule_interval='@daily') as dag:

    send_email_task = PythonOperator(
        task_id='send_email_notification',
        python_callable=send_custom_notification
)

```

3.3.3 Настройка шаблонов писем

В секции [email] файла airflow.cfg можно настроить пользовательские шаблоны:

```

[ini
[email]
email_backend = airflow.utils.email.send_email_smtp
subject_template = /path/to/custom_subject_template.j2
html_content_template = /path/to/custom_html_template.j2

```

3.3.4 Пример пользовательского шаблона HTML

```

jinja2
<h3>Уведомление о выполнении DAG: {{ ti.dag_id }}</h3>
<p><strong>Задача:</strong> {{ ti.task_id }}</p>
<p><strong>Статус:</strong> {{ ti.state }}</p>
<p><strong>Время выполнения:</strong> {{ df_now }}</p>
<p><strong>Ссылка на логи:</strong> <a href="{{ log_url }}">Перейти к логам</a></p>
<p><strong>Базовая ссылка:</strong> {{ base_url }}</p>

{% if ti.state == 'failed' %}
<div style="color: red;">
    <h4>Ошибка выполнения:</h4>
    <pre>{{ exception_html }}</pre>
</div>
{% endif %}

```

Инв. № подл.	Подпись и дата
Изм	Лист

Изм	Лист	№ докум.	Подп.	Дата

3.3.5 Практические рекомендации

- 1) Безопасность учетных данных: Для хранения SMTP пароля рекомендуется использовать секреты Airflow или переменные окружения
- 2) Тестирование подключения: Используйте утилиту airflow config get-value smtp smtp_host для проверки настроек
- 3) Шаблоны по умолчанию: Polyflow включает базовые шаблоны в директории source/df/common/email

3.4 Операции группы безопасности

Через верхнее меню Безопасность доступны формы, относящиеся к безопасности системы.

3.4.1 Список пользователей

Данная форма позволяет найти, просмотреть, добавить, изменить или удалить пользователей системы (Рисунок 3).

Имя	Фамилия	Логин	Email	Активен?	Роль
admin	super	admin	admin@example.com	True	[Admin]

Рисунок 3. Список пользователей

По кнопке «показать запись» по выбранному в таблице пользователю доступна подробная информация и данные по аудиту (Рисунок 4).

Инв. № подл.	Подпись и дата	Инв. № дубл.	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Показать пользователя

Ваши данные

Логин	admin
Активен?	True
Роль	[Admin]
Количество входов	785

Частные данные

Имя	admin
Фамилия	super
Email	admin@example.com

Данные по аудиту

Последний вход	2022-05-11 15:31:39.556015
Не удалось посчитать количество входов	0
Создано	2022-04-20 10:04:09.479935
Кем создано	None
Когда изменено	2022-04-20 10:04:09.479941
Кем изменено	None

[Сбросить пароль](#) [Сброс пароля](#) [←](#)

Рисунок 4. Информация по пользователю
 В случае необходимости данные по пользователю можно отредактировать (Рисунок 5):

Редактировать пользователя

Имя *	admin
Фамилия *	super
Логин *	admin
Активен?	<input checked="" type="checkbox"/>
Email *	admin@example.com
Роль	Admin

[Сохранить](#) [←](#)

Рисунок 5. Редактирование данных пользователя

Для добавления нового пользователя необходимо нажать кнопку «+» и ввести необходимую информацию (Рисунок 6):

Инв. № подл.	Подпись и дата	Подпись и дата	Инв. № подл.
--------------	----------------	----------------	--------------

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

Добавить пользователя

Имя * Имя
Фамилия * Фамилия
Логин * Логин
Активен? Активен?
Удалять пользователя плохая практика, просто сделайте его неактивным
Email * Email
E-mail пользователя. Будет также использоваться для OID аутентификации
Роль * Select value
Роль пользователя, определяющая список его прав
Пароль * Пароль
The user's password for authentication
Подтвердите пароль * Подтвердите пароль
Повторите пароль
Сохранить Отмена

Рисунок 6. Добавление нового пользователя

3.4.2 Список ролей

Данная форма позволяет найти, просмотреть, добавить, изменить или удалить роли системы. Также через форму осуществляется настройка прав ролей (Рисунок 7).

Список ролей

Поиск

Действия

Название Права

Admin

[Can delete on Connections, can read on Connections, can edit on Connections, can create on Connections, can read on DAGs, can edit on DAGs, can delete on DAGs, can read on DAG Runs, can read on Task Instances, can edit on Task Instances, can delete on DAG Runs, can edit on Audit Logs, can read on ImportError, can delete on Pools, can edit on Pools, can create on Pools, can read on Providers, can delete on Variables, can read on Variables, can edit on Variables, can create on Variables, can read on XComs, can read on DAG Code, can read on Configurations, can read on Plugins, can read on Roles, can read on Permissions, can delete on Roles, can edit on Roles, can create on Roles, can read on Users, can edit on Users, can create on Users, can edit on Users, can delete on Users, can edit on Passwords, can read on Passwords, can edit on My Password, can read on My Password, can edit on My Profile, can read on My Profile, can read on User Stats Chart, can read on View Menus, can read on Permission Views, can get on MenuApi, menu access on DAG Runs, can read on Jobs, menu access on Jobs, menu access on Audit Logs, menu access on Variables, can create on Task Instances, can delete on Task Instances, can read on Task Instances, can read on Task Pessimizes, menu access on Task Pessimizes, menu access on Triggers, menu access on Triggers, menu access on Configurations, menu access on Connections, can read on SLA Misses, menu access on SLA Misses, menu access on Plugins, menu access on Providers, menu access on Pools, can create on XComs, menu access on XComs, can list on AppBuilderCodeEditorView, menu access on Security, menu access on List Users, menu access on List Roles, menu access on User's Statistics, menu access on Base Permissions, menu access on View Menus, menu access on Permission Views, menu access on Browse, menu access on DAG Dependencies, menu access on Admin, menu access on DAGs Code Editor, menu access on Docs, menu access on Documentation, can read on DAG Dependencies, can read on Task Logs, can read on Website, can create on DAG Runs]

Public

[Can read on DAGs, can read on DAG Runs, can read on Task Instances, can read on Audit Logs, can read on ImportError, can read on XComs, can read on DAG Code, can read on Plugins, can edit on My Password, can read on My Password, can edit on My Profile, can read on My Profile, can read on User Stats Chart, can read on View Menus, can read on Permission Views, can get on MenuApi, menu access on DAG Runs, can read on Task Logs, can read on Website]

Viewer

[Can read on DAGs, can edit on DAGs, can delete on DAGs, can read on Task Instances, can edit on Task Instances, can delete on Task Instances, can read on DAG Runs, can read on Audit Logs, can edit on My Password, can read on My Password, can edit on My Profile, can read on My Profile, can read on User Stats Chart, can read on View Menus, can read on Permission Views, can get on MenuApi, menu access on DAG Runs, can read on Task Logs, can read on Website]

User

[Can read on DAGs, can edit on DAGs, can delete on DAGs, can read on Task Instances, can edit on Task Instances, can delete on Task Instances, can read on DAG Runs, can read on Audit Logs, can edit on My Password, can read on My Password, can edit on My Profile, can read on My Profile, can read on User Stats Chart, can read on View Menus, can read on Permission Views, can get on MenuApi, menu access on DAG Runs, can read on Task Logs, can read on Website]

Op

[Can delete on Connections, can read on Connections, can edit on Connections, can create on Connections, can read on DAGs, can edit on DAGs, can delete on DAGs, can read on DAG Runs, can read on Task Instances, can edit on Task Instances, can delete on DAG Runs, can edit on Audit Logs, can read on ImportError, can delete on Pools, can edit on Pools, can create on Pools, can read on Providers, can delete on Variables, can read on Variables, can edit on Variables, can create on Variables, can read on XComs, can read on DAG Code, can read on Configurations, can read on Plugins, can edit on My Password, can read on My Password, can edit on My Profile, can read on My Profile, can read on User Stats Chart, can read on View Menus, can read on Permission Views, can get on MenuApi, menu access on DAG Runs, can read on Task Logs, can read on Website]

Количество записей: 5

Рисунок 7. Список ролей

Для выбранной роли доступна информация по назначенным правам (Рисунок 8) и пользователям (Рисунок 9), которым роль доступна.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата	Лист
					46

Показать роль	
Detail	Список пользователей
Название	Viewer
Права	[can read on DAGs, can read on DAG Runs, can read on Task Instances, can read on Audit Logs, can read on ImportError, can read on XComs, can read on DAG Code, can read on Plugins, can edit on My Password, can read on My Password, can edit on My Profile, can read on My Profile, menu access on DAG Runs, can read on Jobs, menu access on Jobs, menu access on Audit Logs, menu access on Task Instances, can read on SLA Misses, menu access on SLA Misses, menu access on Plugins, menu access on Browse, menu access on DAG Dependencies, menu access on Docs, menu access on Documentation, can read on DAG Dependencies, can read on Task Logs, can read on Website]

Рисунок 8. Информация по правам роли

Редактировать роль

Detail	Список пользователей	Количество записей: 1			
+	←				
Имя	Фамилия	Логин	Email	Активен?	Роль
user	one	user1	user1@example.com	True	[User]

Рисунок 9. Информация по пользователям роли

В случае необходимости состав прав роли можно отредактировать (Рисунок 10).

Редактировать роль

Detail Список пользователей

Название *	Viewer
Права	<ul style="list-style-type: none"><input checked="" type="checkbox"/> can read on DAGs<input checked="" type="checkbox"/> can read on DAG Runs<input checked="" type="checkbox"/> can read on Task Instances<input checked="" type="checkbox"/> can read on Audit Logs<input checked="" type="checkbox"/> can read on ImportError<input checked="" type="checkbox"/> can read on XComs<input checked="" type="checkbox"/> can read on DAG Code<input checked="" type="checkbox"/> can read on Plugins<input checked="" type="checkbox"/> can edit on My Password<input checked="" type="checkbox"/> can read on My Password<input checked="" type="checkbox"/> can edit on My Profile<input checked="" type="checkbox"/> can read on My Profile<input checked="" type="checkbox"/> menu access on DAG Runs<input checked="" type="checkbox"/> can read on Jobs<input checked="" type="checkbox"/> menu access on Jobs<input checked="" type="checkbox"/> menu access on Audit Logs<input checked="" type="checkbox"/> menu access on Task Instances<input checked="" type="checkbox"/> can read on SLA Misses<input checked="" type="checkbox"/> menu access on SLA Misses<input checked="" type="checkbox"/> menu access on Plugins<input checked="" type="checkbox"/> menu access on Browse<input checked="" type="checkbox"/> menu access on DAG Dependencies<input checked="" type="checkbox"/> menu access on Docs<input checked="" type="checkbox"/> menu access on Documentation<input checked="" type="checkbox"/> can read on DAG Dependencies<input checked="" type="checkbox"/> can read on Task Logs<input checked="" type="checkbox"/> can read on Website

Сохранить  

Рисунок 10. Редактирование роли

Также через форму управления ролями доступно добавление новых пользователей той или иной роли.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

3.4.3 Статистика пользователя

Форма отображает информацию со статистикой по пользователям (Рисунок 11).

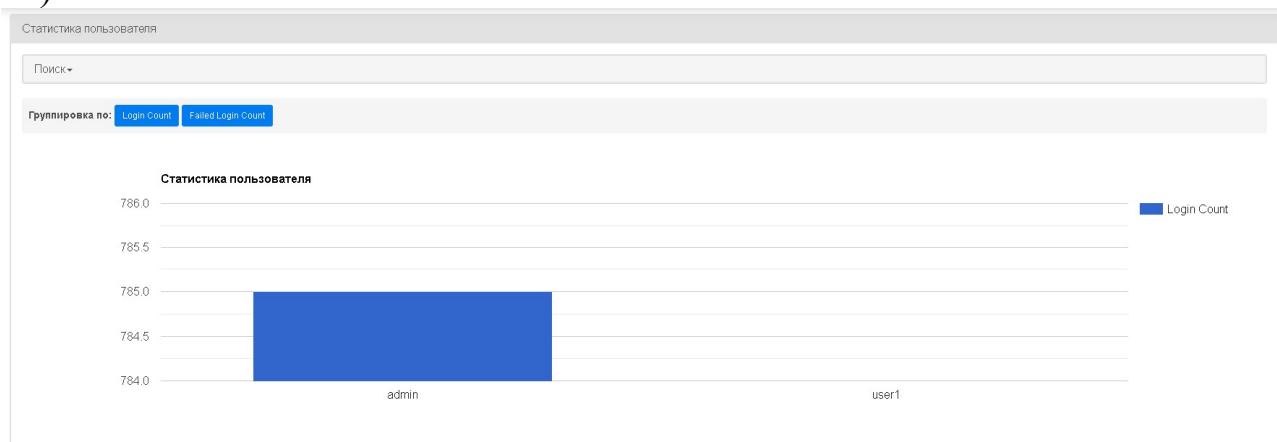


Рисунок 11. Статистика пользователя

3.4.4 Основные разрешения

Форма отображает список базовых прав доступа (элементарных действий над ресурсами системы, такими как: DAG, запуски DAG, задачи и подключения) (Рисунок 12).

Список базовых прав доступа	
Поиск	
<input type="button" value="←"/>	
Количество записей: 8	
Название	
can_delete	
can_read	
can_edit	
can_create	
can_get	
menu_access	
can_list	
menu_acccess	

Рисунок 12. Список базовых прав доступа

3.4.5 Отображения/меню

Форма отображает список отображений/меню, на которые возможна выдача базовых прав доступа (Рисунок 13).

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.

Изм	Лист	№ докум.	Подп.	Дата

Список меню	
Поиск	▼
<	1 2 3 4 5 6 7 >
размер страницы	▼
Название	
Admin	
Airflow	
AppBuilderCodeEditorView	
Audit Logs	
AuthDBView	
AutocompleteView	
Base Permissions	
Browse	
Configurations	
Connections	
DAG branch_test	
DAG Code	
DAG Dependencies	
DagDependenciesView	
DAG df_airflow_log_cleanup	
DAG df_backup_environment	
DAG df_datadb_populate	
DAG df_datadb_upgrade	
DAG df_example_basic_elt_workflow	
DAG df_example_basic_sql	
DAG df_example_batch_sql	
DAG df_example_create_sequence	
DAG df_example_dagmon_observer	
DAG df_example_dagmon_subject	
DAG df_example_data_move	

Рисунок 13. Список меню

3.4.6 Право на просмотр/меню

Форма отображает список прав на просмотр/меню (Рисунок 14).

3.4.6 Право на просмотр/меню

Форма отображает список прав на просмотр/меню (Рисунок 14).

Список прав на просмотр/меню	
Полкскв	
< < 1 2 3 4 5 6 7 > размер страницы < >	
разрешение	Отображение/меню
can_delete	Connections
can_read	Connections
can_edit	Connections
can_create	Connections
can_read	DAGs
can_edit	DAGs
can_delete	DAGs
can_read	DAG Runs
can_read	Task Instances
can_edit	Task Instances
can_delete	DAG Runs
can_edit	DAG Runs
can_read	Audit Logs
can_read	ImportError
can_delete	Pools
can_read	Pools
can_edit	Pools
can_create	Pools
can_read	Providers
can_delete	Variables
can_read	Variables
can_edit	Variables
can_create	Variables
can_read	XComs
can_read	DAG Code

Рисунок 14. Список прав на просмотр/меню

3.5 Операции группы Admin

К операциям администрирование Polyflow относятся: управление переменными, подключениями, провайдерами, плагинами.

Меню Admin содержит пункты:

3.5.1 Variables

Содержит ссылку на страницу с переменными, доступными для использования в DAG (Рисунок 15).

Выберите файл	Файл не выбран	 Импорт переменных
List Variable		
Полкскв		
 Действия		

Рисунок 15. Переменные

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

Форма позволяет просматривать, добавлять, редактировать (Рисунок 16), импортировать и экспорттировать переменные (Рисунок 17).

Key	logical_date
Val	31.03.2022
Description	Description

Сохранить Отмена

Рисунок 16. Редактирование переменной

logical_date	31.03.2022	Description	True
--------------	------------	-------------	------

Удалить Экспорт

Рисунок 17. Действия над переменными

3.5.2 Configuration

Содержит ссылку на страницу с информацией о текущей конфигурации. Доступность страницы и информации может быть задана настройками.

3.5.3 Connections

Предназначен для настройки подключений (Рисунок 18).

Инв. № подл.	Подпись и дата	Подпись и дата	Инв. № дубл.	Взам. инв. №	Взам. инв. №
--------------	----------------	----------------	--------------	--------------	--------------

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

List Connection							
Поле							
	Conn Id	Conn Type	Description	Host	Port	Is Encrypted	Is Extra Encrypted
<input type="checkbox"/>	<input checked="" type="checkbox"/> aws_default	aws				False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> azure_batch_default	azure_batch				False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> azure_cosmos_default	azure_cosmos				False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> azure_data_explorer_default	azure_data_explorer		https://<CLUSTER>.kusto.windows.net		False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> azure_data_lake_default	azure_data_lake				False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> azure_default	azure				False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> cassandra_a_default	cassandra	cassandra		9042	False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> databricks_default	databricks	localhost			False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> dingding_default	http				False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> drill_default	drill	localhost		8047	False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> druid_broker_default	druid	druid-broker		8082	False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> druid_ingest_default	druid	druid-overlord		8081	False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> elasticsearch_default	elasticsearch	localhost		9200	False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> emr_default	emr				False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> facebook_default	facebook_social				False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> fs_default	fs				False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> google_cloud_default	google_cloud_platform				False	False
<input type="checkbox"/>	<input checked="" type="checkbox"/> hive_cli_default	hive_cli	localhost		10000	False	True
<input type="checkbox"/>	<input checked="" type="checkbox"/> hiveserver2_default	hiveserver2	localhost		10000	False	False

Рисунок 18. Подключения

Форма позволяет просматривать, добавлять, редактировать подключения (Рисунок 19).

Edit Connection

Идентификатор соединения *	<input type="text" value="testpgsql"/>
Тип подключения *	<input type="text" value="Postgres"/>
Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.	
Описание	<input type="text"/>
Хост	<input type="text" value="172.24.0.41"/>
Схема	<input type="text" value="appdb"/>
Логин	<input type="text" value="dwh"/>
Пароль	<input type="text"/>
Порт	<input type="text" value="5432"/>
Дополнительно	<input "dwh"}"="" schema":="" type="text" value="{"/>

Рисунок 19. Редактирование подключения

3.5.4 Plugins

Предназначен для просмотра информации о плагинах (Рисунок 20, Рисунок 21, Рисунок 22).

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Airflow Plugins	
1. datflow_api_plugin	
Атрибут	Значение
hooks	[]
executors	[]
macros	[]
admin_views	[]
flask_blueprints	[<flask.blueprints.Blueprint object at 0x7fa3efe48760>]
menu_links	[]
appbuilder_views	[{'category': 'Docs', 'name': 'Dataflow API OpenAPI', 'view': <view.AppView object at 0x7fa3efe52310>}]
appbuilder_menu_items	[{'category': 'Docs', 'name': 'about Dataflow API', 'href': 'https://gitlab-ocv.polymedia.ru/system/wiki'}]
global_operator_extra_links	[]
operator_extra_links	[]
source	\$/PLUGINS_FOLDER/views/dataflow_api/view.py

Рисунок 20. Информация о плагинах

2. setup	
Атрибут	Значение
hooks	[]
executors	[]
macros	[]
admin_views	[]
flask_blueprints	[]
menu_links	[]
appbuilder_views	[]
appbuilder_menu_items	[]
global_operator_extra_links	[]
operator_extra_links	[]
source	<code>\$PLUGINS_FOLDER/common/setup.py</code>

Рисунок 21. Информация о плагинах (продолжение)

3. editor_plugin	
Атрибут	Значение
hooks	[]
executors	[]
macros	[]
admin_views	[]
flask_blueprints	[<flask.blueprints.Blueprint object at 0x7fa3efdfcfa0>]
menu_links	[]
appbuilder_views	[{'category': 'Admin', 'name': 'DAGs Code Editor', 'view': <airflow_code_editor.app_builder_view.AppBuilderCodeEditorView object at 0x7fa3efef1f7f0>}]
appbuilder_menu_items	[]
global_operator_extra_links	[]
operator_extra_links	[]
source	airflow_code_editor==5.1.5: EntryPoint(name='airflow_code_editor', value='airflow_code_editor.airflow_code_editor:CodeEditorPlugin', group='airflow.plugins')

Рисунок 22. Информация о плагинах (продолжение 2)

3.5.5 Providers

Предназначен для просмотр информации о установленных провайдерах (Рисунок 23).

3. editor_plugin				
Атрибут	Значение			
hooks	[]			
executors	[]			
macros	[]			
admin_views	[]			
flask_blueprints	[<flask.blueprints.Blueprint object at 0x7fa3efdfcfa0>]			
menu_links	[]			
appbuilder_views	[{'category': 'Admin', 'name': 'DAGs Code Editor', 'view': <airflow_code_editor.app_builder_view.AppBuilderCodeEditorView object at 0x7fa3efef1f7f0>}]			
appbuilder_menu_items	[]			
global_operator_extra_links	[]			
operator_extra_links	[]			
source	airflow_code_editor==5.1.5; EntryPoint(name='airflow_code_editor', value='airflow_code_editor.airflow_code_editor:CodeEditorPlugin', group='airflow.plugins')			

Рисунок 22. Информация о плагинах (продолжение 2)

3.5.5 Providers

Предназначен для просмотр информации о установленных провайдерах (Рисунок 23).

Providers		
Имя пакета	Версия	Описание
apache-airflow-providers-cncf-kubernetes	3.0.0	Kubernetes
apache-airflow-providers-docker	2.5.2	Docker
apache-airflow-providers-ftp	2.1.2	File Transfer Protocol (FTP)
apache-airflow-providers-http	2.1.2	Hypertext Transfer Protocol (HTTP)
apache-airflow-providers-imap	2.2.3	Internet Message Access Protocol (IMAP)
apache-airflow-providers-microsoft-mssql	2.1.3	Microsoft SQL Server (MSSQL)
apache-airflow-providers-postgres	4.1.0	PostgreSQL
apache-airflow-providers-sqlite	2.1.3	SQLite

Рисунок 23. Провайдеры

3.5.6 Pools

Содержит ссылку на страницу с информацией о пулах (Рисунок 24).

List Pool				
Количество записей: 1				
Действия		Pool	Slots	Running Slots
<input type="checkbox"/>	<input checked="" type="checkbox"/>	default_pool	128	0

Рисунок 24. Пул процессов

Форма позволяет просматривать, добавлять, редактировать данные по пулам (Рисунок 25).

Edit Pool	
Pool *	default_pool
Slots	128
Description	Default pool
<input style="width: 100px; height: 25px; margin-right: 10px;" type="button" value="Сохранить"/> <input style="width: 100px; height: 25px;" type="button" value="Отмена"/>	

Рисунок 25. Редактирование данных пула процессов

3.5.6.1 Масштабирование и управление задачами через пулы воркеров

Архитектура Polyflow предоставляет мощные механизмы для распределения задач между несколькими воркерами (исполнителями), что позволяет эффективно масштабировать вычислительную мощность и управлять нагрузкой системы.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

3.5.6.2 Механизм выполнения задач

Polyflow поддерживает различные типы Executor - компонентов, отвечающих за распределение и выполнение задач:

Таблица 6. Типы Executor

Executor	Описание	Преимущества
LocalExecutor	Выполняет задачи в параллельных процессах на одном сервере	Простая настройка, подходит для средних нагрузок
CeleryExecutor	Распределяет задачи между несколькими воркерами через брокер сообщений (Redis/RabbitMQ)	Горизонтальное масштабирование, отказоустойчивость
KubernetesExecutor	Запускает каждую задачу в отдельном Pod в Kubernetes кластере	Максимальная изоляция, эластичное масштабирование

3.5.6.3 Настройка пулов воркеров (Pools)

Пул воркеров - это механизм ограничения параллельного выполнения определенных типов задач для предотвращения перегрузки системы и управления ресурсами.

Создание и настройка пулов через веб-интерфейс:

- 1) Перейдите в меню Admin → Pools
- 2) Нажмите кнопку "+" для создания нового пула
- 3) Задайте параметры:
 - Pool Name: Уникальное имя пула (например, heavy_queries_pool)
 - Slots: Количество параллельных задач в пуле (например, 3 для ресурсоемких операций)
 - Description: Описание назначения пула

Пример использования пула в DAG:

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

```

#python

from airflow import DAG

from airflow.operators.docker_operator import DockerOperator

from datetime import datetime


default_args = {
    'owner': 'data_team',
    'start_date': datetime(2024, 1, 1),
}

with DAG('example_pool_dag',
         default_args=default_args,
         schedule_interval='@daily') as dag:

    # Задача с ограничением через пул
    heavy_etl_task = DockerOperator(
        task_id='heavy_etl_processing',
        image='df_operator:latest',
        command='python /app/run_heavy_etl.py',
        pool='heavy_queries_pool', # Использование специального пула
        pool_slots=1, # Количество слотов, занимаемых задачей
        environment_variables={
            'AF_DWH_DB_SERVER': '{{ var.value.get("db_host") }}',
            'AF_DWH_DB_NAME': 'dwh_db'
        }
    )

    # Обычная задача без ограничений
    light_transform_task = DockerOperator(
        task_id='light_data_transform',
        image='df_operator:latest',
        command='python /app/run_light_transform.py',
        # Не указан pool - выполняется в default_pool
    )

```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

3.5.6.4 Практические сценарии использования пулов

Таблица 7. Типы Executor

Сценарий	Настройка пула	Преимущество
Ограничение тяжелых ETL-задач	heavy_etl_pool с 2-3 слотами	Предотвращение перегрузки БД при параллельном выполнении ресурсоемких запросов
Контроль API-вызовов	external_api_pool с 5-10 слотами	Соблюдение лимитов RPS (requests per second) внешних API
Приоритизация задач	critical_pool (10 слотов), normal_pool (20 слотов)	Гарантированная производительность для критически важных процессов

3.5.6.5 Мониторинг и управление

- Просмотр загрузки пулов: В интерфейсе Polyflow в разделе Pools отображается количество занятых/свободных слотов
- Динамическое изменение: Размер пула можно изменять в runtime без перезапуска системы
- Интеграция с мониторингом: Метрики использования пулов доступны для интеграции с Prometheus/Grafana

3.5.6.6 Рекомендации по настройке

- Размер default_pool: Увеличьте количество слотов в пуле по умолчанию в соответствии с ресурсами сервера
- Балансировка нагрузки: Создавайте специализированные пулы для разных типов задач (ETL, API-вызовы, файловые операции)
- Резервирование ресурсов: Оставляйте 20-30% слотов свободными для внеплановых задач и обработки пиковой нагрузки.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

3.5.6.7 Масштабирование в облачных средах

Механизмы пулов воркеров Polyflow являются фундаментом для эластичного горизонтального масштабирования в облачных средах. Для автоматизации этого процесса в продакшн-развертываниях рекомендуется использовать оркестраторы, такие как Kubernetes, в связке с механизмом Horizontal Pod Autoscaler (HPA). Данная конфигурация позволяет автоматически увеличивать количество подов-воркеров при росте нагрузки и уменьшать его при его снижении, обеспечивая отказоустойчивость и экономическую эффективность.

3.5.7 XComs

Содержит ссылку на страницу с информацией по XComs (Рисунок 26).

List XComs						
Поиск:						
Количество записей: 2815						
	Key	Value	Timestamp	Logical Date	Task Id	Dag Id
□	return_value	Всего загружено записей 6 в user.test_const.	2022-04-20, 13:52:06	2022-04-20, 13:51:56	df_example_extract_excel_const.extract	df_example_extract_excel_const
□	return_value	Выполняется слияние данных в таблицу <rule>	2022-04-20, 13:08:41	2022-04-20, 13:08:30	merge	df_datadb_populate
□	return_value	Выполнение запроса SQL завершено.	2022-04-20, 13:08:16	2022-04-20, 13:08:04	callable_common	df_datadb_upgrade
□	return_value	INFO [alembic.runtime.migration] Will assume transactional DDL.	2022-04-20, 13:08:09	2022-04-20, 13:08:04	ddl	df_datadb_upgrade
□	return_value	Выполнение запроса SQL завершено.	2022-04-20, 13:08:23	2022-04-20, 13:08:04	callable_json	df_datadb_upgrade
□	return_value	Выполнение запроса SQL завершено.	2022-04-20, 13:08:21	2022-04-20, 13:08:04	callable_rules	df_datadb_upgrade
□	return_value	'df_example_dagmon_subject' начал свою работу	2022-04-28, 01:24:54	2021-01-01, 03:00:00	df_example_dagmon_subject.bash_task1	df_example_dagmon_subject
□	trigger_run_id	manual_2022-04-27T08:59:43.140306+00:00	2022-04-27, 11:59:43	2021-01-01, 03:00:00	df_example_monitoring_observer.subject_trigger	df_example_monitoring_observer
□	trigger_execution_date_iso	2022-04-27T22:24:54.912091+00:00	2022-04-28, 01:24:55	2021-01-01, 03:00:00	df_example_dagmon_subject.observer_trigger	df_example_dagmon_subject
□	trigger_execution_date_iso	2022-04-27T08:59:43.203879+00:00	2022-04-27, 11:59:43	2021-01-01, 03:00:00	df_example_monitoring_observer.observer_trigger	df_example_monitoring_observer
□	trigger_run_id	manual_2022-04-27T22:24:54.912091+00:00	2022-04-28, 01:24:55	2021-01-01, 03:00:00	df_example_dagmon_subject.observer_trigger	df_example_dagmon_subject
□	trigger_run_id	manual_2022-04-27T08:59:43.203879+00:00	2022-04-27, 11:59:43	2021-01-01, 03:00:00	df_example_monitoring_observer.observer_trigger	df_example_monitoring_observer
□	return_value	'df_example_dagmon_subject' завершил свою работу	2022-04-28, 01:25:08	2021-01-01, 03:00:00	df_example_dagmon_subject.bash_task2	df_example_dagmon_subject
□	return_value	df_example_monitoring_subject начал свою работу	2022-04-27, 11:50:11	2021-01-01, 03:00:00	df_example_monitoring_subject.bash_task1	df_example_monitoring_subject
□	trigger_execution_date_iso	2022-04-27T08:59:43.140306+00:00	2022-04-27, 11:59:43	2021-01-01, 03:00:00	df_example_monitoring_observer.subject_trigger	df_example_monitoring_observer

Рисунок 26. XComs

3.5.8 DAGs Code Editor

Содержит ссылку на страницу редактора DAG (Рисунок 27, Рисунок 28).

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.

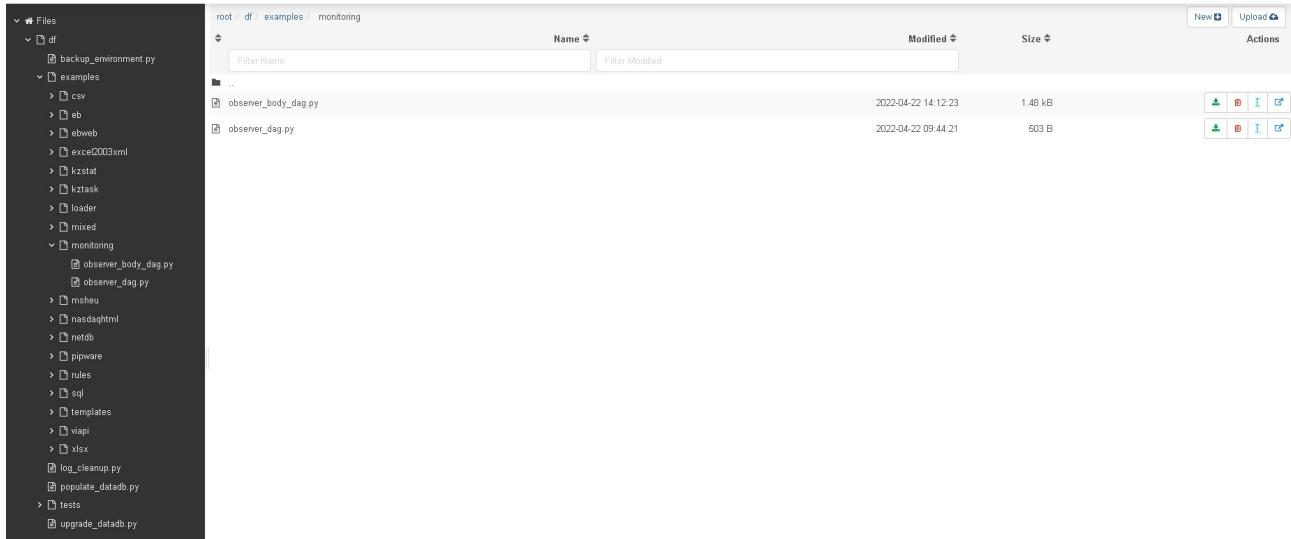


Рисунок 27. Редактор DAG

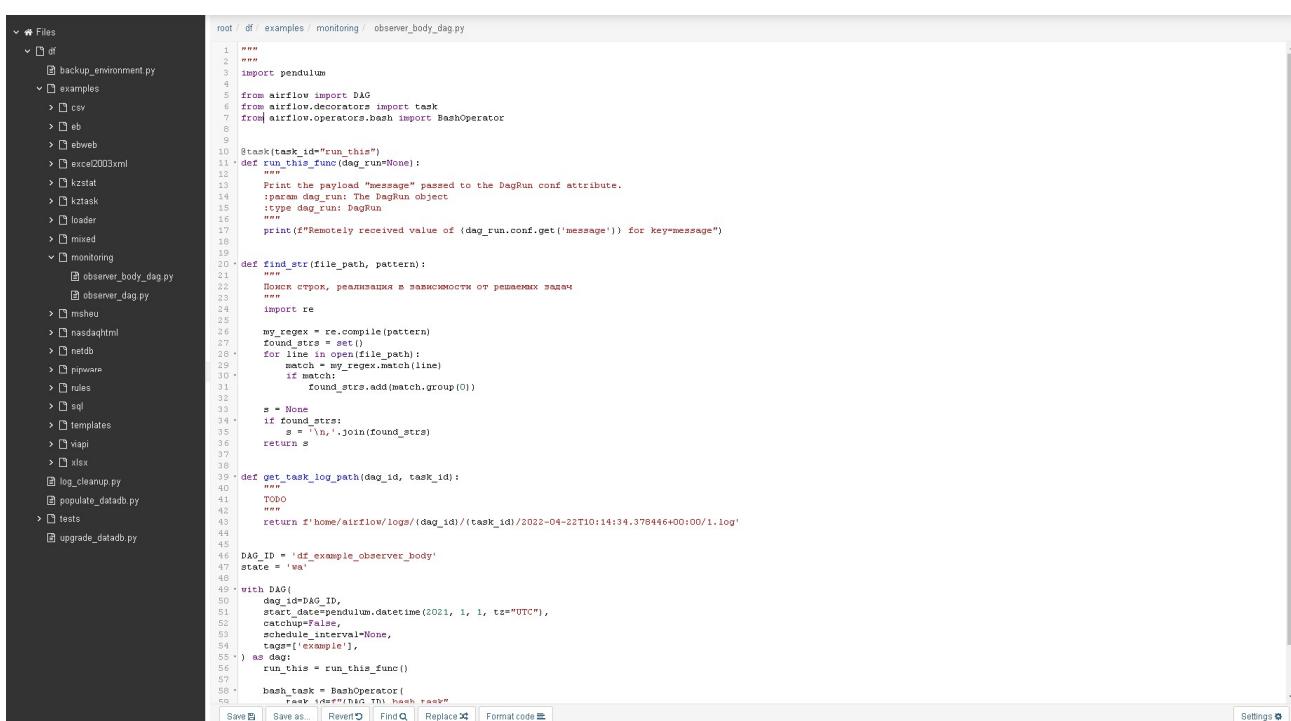


Рисунок 28. Редактор DAG с открытым файлом

3.6 Политики безопасности и шифрования

3.6.1 Шифрование передаваемых данных

Polyflow обеспечивает многоуровневое шифрование данных:

Уровень 1: Веб-интерфейс

- HTTPS с SSL/TLS сертификатами

- Шифрование аутентификационных данных
- Защита сессий пользователей

Уровень 2: Внутренние коммуникации

- Шифрование трафика между компонентами

Уровень 3: Внешние подключения

- SSL/TLS для подключений к СУБД
- Шифрование API-вызовов к внешним системам
- Защита данных в транзите при работе с облачными провайдерами

3.6.2 Рекомендации по настройке

- 1) Сертификаты: Используйте доверенные SSL-сертификаты
- 2) Алгоритмы: Применяйте современные криптографические алгоритмы
- 3) Обновление: Регулярно обновляйте сертификаты и ключи
- 4) Мониторинг: Контролируйте состояние шифрованных соединений

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

4 Аварийные ситуации

Для Системы определены следующие режимы функционирования:

- штатный;
- аварийный.

Аварийный режим функционирования Системы используется при отказе одного или нескольких компонент программного и (или) технического обеспечения.

При переходе в аварийный режим в Системе предусмотрено формирование соответствующего информационного сообщения.

После выдачи сообщения, администратору необходимо выполнить комплекс мероприятий по устранению причины перехода Системы в аварийный режим.

При работе с АИС могут возникнуть следующие неисправности, приводящие к аварийным ситуациям:

- Превышение нагрузки на АИС. В этом случае необходимо ограничить количество тяжело-нагруженных процессов или общее их количество;
- Недостаток свободной оперативной памяти на сервере. В этом случае необходимо ограничить ресурсы для контейнера.
- Другие неисправности. В случае нарушения технологического процесса или при длительных отказах технических средств администратор системы обязан сообщить о возникшей проблеме в службу технической поддержки, провести диагностику работы Системы, определить вероятную причину неисправности и передать лог-файлы из соответствующего docker-контейнера. Чтобы связаться с службой поддержки необходимо сообщить о возникшей неисправности по электронному адресу: support@polyanalitika.ru.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

5 Рекомендации по освоению

Основным источником информации, используемым при освоении Системы, является данное руководство.

Начинать работу с Системой следует со знакомства с разделами руководства «Подготовка к работе», «Описание операций».

Для обеспечения успешной работы пользователям необходимо обладать основными навыками работы с веб-приложениями, опубликованными в сети Интернет.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата
--------------	----------------	--------------	--------------	----------------

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

РЭ

Лист регистрации изменений

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм	Лист	№ докум.	Подп.	Дата

РЭ

Лист

64