Polyflow. Описание функциональных возможностей и архитектуры решения

Содержание

Polyflow

Система управления данными Polyflow (далее — Система) – это решение для управления корпоративными хранилищами данных (КХД).
Модуль представляет собой сервис оркестровки, сбора и обработки разнородных данных в хранилищах произвольной архитектуры. Polyflow позволяет управлять следующей функциональностью:

- 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.

Определения и сокращения Polyflow

Определения и сокращения Polyflow представлены в Таблица 1.

Таблица 1. Определения и сокращения Polyflow

Термин/Сокращение	Определение
Хранилище данных (англ. Content Repository, Data Warehouse, DWH)	Предметно-ориентированная информационная база данных, сочетающая в себе функции системы управления версиями, поисковой машины и СУБД
КХД	Корпоративное хранилище данных
Система управления данными Polyflow	Сервис оркестровки сбора и обработки разнородных данных хранилища произвольной архитектуры
Polyflow	Краткое наименование программного обеспечения «Система управления данными Polyflow»
API	«Application Programming Interface», интерфейс программирования приложений, программный интерфейс приложения
ККД	Контроль качества данных – оценка соответствия данных заданным критериям

Архитектура решения

Логическая архитектура (функциональные модули)

Система управления данными Polyflow построена по модульному принципу, где каждый компонент отвечает за определенный функциональный набор задач. Взаимодействие между модулями определяет логику работы системы по сбору, обработке и контролю качества данных.

Состав компонент

Система включает в себя следующие модули:

1. Планировщик/оркестратор
2. Модули расширения функционала
3. Модуль ETL (загрузка и обработка)
4. Модуль управления ККД
5. Модуль управления метаданными

Планировщик/оркестратор

Выполняет функции планирования, мониторинга и запуска процессов системы, синхронизации и ведения истории их выполнения.

Модули расширения функционала

Дополнительные модули, упрощающие взаимодействие с планировщиком и оркестратором, настройку процессов, а также взаимодействие компонентов системы.

Модуль ETL (загрузка и обработка)

Модуль, выполняющий функции извлечения, трансформации и загрузки данных из различных источников.

Модуль управления ККД

Модуль, решающий задачи контроля качества данных.

Модуль управления метаданными

Модуль, предоставляющий информацию о моделях для других компонентов системы.

Основные взаимодействия

На Рисунок 1 представлена схема решения и основные взаимодействия между его частями и внешними системами:

1. Запрос на скачивание файла из внешнего хранилища файлов;
2. Подготовка файла для скачивания;
3. Скачивание файла;
4. Инициирование извлечения данных из внешнего источника через API;
5. Ответ внешней системы: код ошибки или получение данных;
6. Инициирование извлечения данных из внешней базы данных;
7. Ответ внешней системы: код ошибки или получение данных;
8. Инициирование и загрузка данных в хранилище;
9. Инициирование запуска правил ККД на стороне хранилища;
10. Ответ с результатом выполнения правил ККД;
11. Взаимодействие с метаданными хранилища;

12. Взаимодействие сервисов планировщика и оркестратора с модулями расширения;
13. Взаимодействие модулей расширения с модулем ETL;
14. Взаимодействие модулей расширения с модулем управления ККД;
15. Взаимодействие модулей расширения с модулем управления метаданными;
16. Получение информации о моделях, используемых в ETL (описание сущностей-источников и получателей);
17. Получение информации о моделях, используемых в ККД (описание правил).

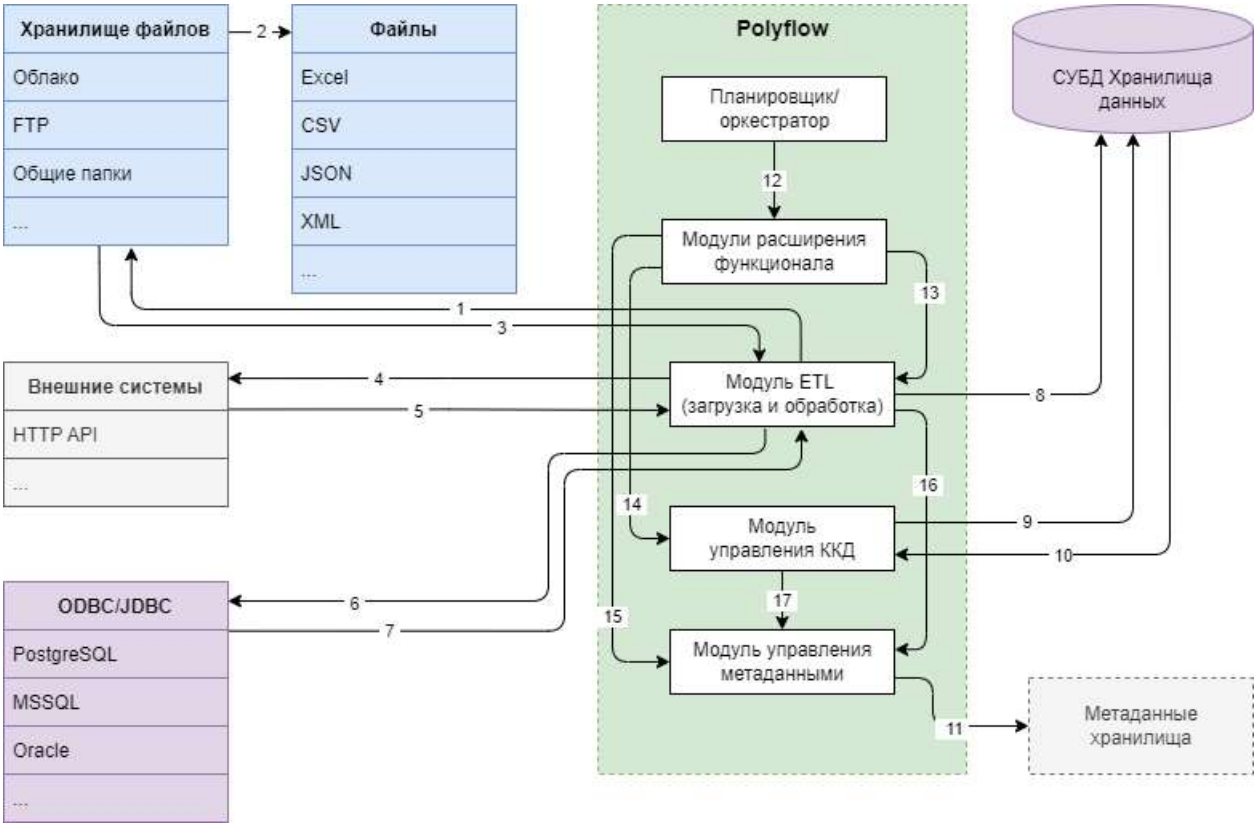

Рисунок 1. Архитектура решения

Физическая архитектура (развертывание)

В основе физической реализации Polyflow лежит контейнеризация на базе Docker. Система состоит из набора сервисов, оркестрируемых с помощью Docker Compose. Каждый функциональный модуль логической архитектуры реализован в виде одного или нескольких взаимодействующих контейнеров.

Такой подход обеспечивает:

- Масштабируемость — возможность горизонтального масштабирования отдельных компонентов;
- Изолированность — выполнение задач ETL (загрузки и обработки) и ККД в отдельных контейнерах;
- Воспроизводимость — идентичность окружений на разных стендах.

Состав сервисов Docker

Система развертывается как набор взаимодействующих Docker-контейнеров, объединенных общей сетью app-tier. Основные сервисы (веб-сервер (webserver), планировщик (scheduler),

инициатор (triggerer)) используют локальный образ df_airflow:latest, а для выполнения задач ETL и ККД динамически создаются контейнеры на основе образа df_operator:latest.

Иерархия образов

При установке системы у клиента формируются два основных образа, которые собираются индивидуально на основе предоставленных базовых образов Таблица 2:

Таблица 2. Состав Docker-образов Polyflow

Образ	Базовый образ	Назначение
df_airflow:latest	registry.polyanalitika.ru/r5/df_airflow:<version>	Используется для сервисов оркестрации — веб-сервера, планировщика и инициатора. Содержит компоненты Airflow и модули расширения Polyflow.
df_operator:latest	registry.polyanalitika.ru/r5/df_operator: <version>	Используется для динамически создаваемых контейнеров задач (Task Container), выполняющих задачи ETL (загрузки и обработки) и контроля качества данных. Включает необходимые драйверы, библиотеки и утилиты для подключения к различным источникам данных.

Таблица 3. Состав Docker-сервисов Polyflow

Сервис	Имя контейнера	Базовый образ	Назначение
postgres	df-database	registry.polyanalitika.ru/r5/r5_postgres13:1.0	Хранение метаданных Polyflow (графов задач (DAG), подключений (connections), переменных (variables), история выполнения).
веб-сервер (webserver)	df-webserver	df_airflow:latest	Веб-интерфейс Polyflow для мониторинга и управления процессами. Доступен по HTTPS на порту 8080.
планировщик (scheduler)	df-scheduler	df_airflow:latest	Планировщик задач — отслеживает расписания графов задач (DAG) и инициирует выполнение задач.
инициатор (triggerer)	df-triggerer	df_airflow:latest	Обработчик отложенных (deferred) задач — отслеживает и возобновляет задачи, ожидающие внешних событий.
контейнер задач (task container)	(динамические имена)	df_operator:latest	Временные контейнеры, создаваемые для выполнения конкретных задач ETL (загрузки и обработки) или ККД.

Ключевые особенности конфигурации

- Индивидуальная сборка: при установке у клиента образы df_airflow:latest и df_operator:latest собираются индивидуально на основе предоставленных базовых образов, что позволяет учитывать особенности конкретного окружения.
- Разделение ответственности: образ df_airflow отвечает за оркестрацию и управление, а образ df_operator — за выполнение задач по обработке данных. Это обеспечивает изоляцию и безопасность, так как код операторов выполняется в отдельных контейнерах.

- Монтирование Docker.sock: все сервисы имеют доступ к сокету Docker на хосте (/var/run/docker.sock), что позволяет планировщику динамически создавать контейнеры на основе образа df_operator:latest для выполнения задач.
- Внешние тома: для сохранения данных между перезапусками используются внешние тома Docker:
    - df_data — данные PostgreSQL;
    - df_logs — логи выполнения;
    - df_projects и df_workspace — проекты и рабочее пространство Polyflow.
- Секреты: для обеспечения безопасности подключения по HTTPS используются сертификаты, передаваемые в контейнер webserver через Docker secrets.

На рисунке Рисунок 2 представлена схема физической архитектуры с указанием сетевых взаимодействий и портов.

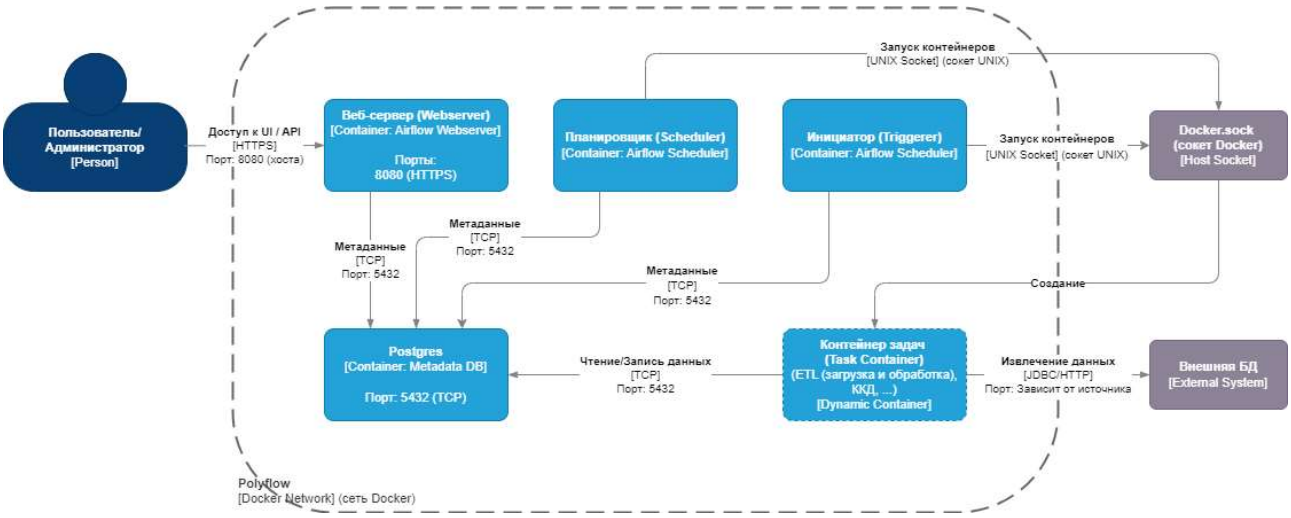

Рисунок 2. Схема взаимодействия компонентов Polyflow

Сетевые взаимодействия компонентов

В таблице Таблица 4 приведено детальное описание взаимодействия между контейнерами системы и внешними компонентами с указанием протоколов и портов.

Таблица 4. Взаимодействие компонентов при развертывании

Компонент А	Компонент Б	Направление	Протокол	Порт (Компонента Б)	Описание взаимодействия
Пользователь /Система	Веб-сервер (Webserver)	→	HTTP	8080 (хоста)	Доступ к веб-интерфейсу Polyflow.
Веб-сервер (Webserver)	Postgres	→	TCP	5432	Чтение/запись метаданных Polyflow (Запуски задач (DAG runs), подключения (connections), переменные (variables)).
Планировщик (Scheduler)	Postgres	→	TCP	5432	Чтение/запись метаданных Polyflow, обновление статусов задач, постановка задач в очередь.
Инициатор (Triggerer)	Postgres	→	TCP	5432	Чтение/запись метаданных для отложенных (deferred) задач, обновление их статусов.

Компонент А	Компонент Б	Направление	Протокол	Порт (Компонента Б)	Описание взаимодействия
Веб-сервер (Webserver)	Планировщик (Scheduler) / Инициатор (Triggerer)	↔	(внутреннее)	-	Прямой связи нет, координация осуществляется через базу данных Postgres.
Планировщик (Scheduler) / Веб-сервер (Webserver) / Инициатор (Triggerer)	Docker.sock (сокет Docker)	→	(UNIX socket)	-	Запуск и управление контейнерами с задачами (ETL (загрузка и обработка), ККД и др.). Важно: в вашей конфигурации доступ к сокету есть у всех сервисов, так как они используют общий x-df-common.
Контейнер задач (Task Container)	Postgres	→	TCP	5432	Загрузка или извлечение данных в/из корпоративного хранилища данных (КХД).
Контейнер задач (Task Container)	Внешние системы	→	HTTP / JDBC / и др.	Зависит от источника	Извлечение данных из файловых хранилищ, API, БД (см. функциональное описание Polyflow).