

**Руководство администратора
для системы
«Sym4.Treasury Solutions»**

Оглавление

Общее описание	3
Open Auth Server	3
Процесс установки	4
Среда установки	4
Стандартный процесс установки сервисов	4
Сервис управления бизнес процессами	4
Kubernetes Spark Operator	5
Apache Hive	5
Apache Druid	5

Общее описание

Система Treasure Solution не является монолитной/одной системой. Это набор различных сервисов, взаимодействующих как между собой, так и с UI через сервис Facade.

Open Auth Server

Treasure Solution использует существующий сервер OAuth. Этот сервер должен быть развернут и доступен до начала установки Treasure Solution.

Примеры OAuth серверов:

[OAuth 2.1 Authorization Server :: Spring Security](#)

[Documentation - Keycloak](#)

Процесс установки

Среда установки

На данный момент времени в качестве среды установки выбран Kubernetes. Весь описанный процесс установки сделан исходя из «чистого» Kubernetes. Соответственно Treasure Solution может быть установлен как на чистый Kubernetes, так и на любой коммерческий дистрибутив Kubernetes (например OpenShift).

Стандартный процесс установки сервисов

Для каждого сервиса в исходных кодах приложена отдельная папка/проект с манифестами Kubernetes установки.

Сервисы могут быть установлены через настроенный процесс доставки приложения или путем выполнения команды `kubectl -n namespace apply -f application_dir`, где `application_dir` это папка с именем сервиса в проекте `xxxx-manifests`. Проекты с манифестами для разворачивания в Kubernetes созданы для каждого сервиса.

Некоторым из сервисов нужна для работы база данных PostgreSQL. Установка базы PostgreSQL для каждого сервиса описана в манифестах установки.

Если в окружении клиента не допускается установка баз данных в среде Kubernetes, то этап установки PostgreSQL в Kubernetes может быть удален из манифестов.

Параметры подключения сервиса к базе прописываются через переменные окружения и могут быть гибко настроены на любой вариант взаимодействия с базами данных принятый у заказчика.

Сервис управления бизнес-процессами

Все сервисы Treasure Solution управляются через внешний сервис. На данный момент времени в качестве сервиса управления бизнес-процессами выбран [Apache Airflow](#)

Установка Apache Airflow в среде Kubernetes описана на сайте самого проекта Apache Airflow [Kubernetes — Airflow 3.0.6 Documentation](#)

После установки необходимо настроить [kubernetes/git-sync: A sidecar app which clones a git repo and keeps it in sync with the upstream.](#)

И указать GitSync брать данные из включенного в поставку проекта `alm-airflow-dags`

Если у заказчика нет в наличии Git, то содержимое проекта `alm-airflow-dags` можно скопировать в папку `dags` сервиса Apache Airflow

Kubernetes Spark Operator

Все расчеты делаются с помощью Apache Spark в среде Kubernetes. Процесс настройки Apache Spark как часть инфраструктуры Kubernetes описан на сайте проекта [spark-operator | Kubernetes operator for managing the lifecycle of Apache Spark applications on Kubernetes](#) или [kubeflow/spark-operator](#)

Apache Hive

Сервисы расчетов сохраняют результаты в Apache Hive. Apache Hive должен быть установлен отдельно.

В проекте есть manifest файлы для установки Hive в среде Kubernetes.

Apache hive является «абстракцией» над хранимыми данными. Соответственно место хранения данных Apache Hive также должно быть настроено.

В проекте есть пример как установить и связать Apache Hive с сервисом хранения данных S3.

Apache Druid

Часть данных проекта Treasure Solution необходимо хранить и предоставлять быстро. Для этих целей используется [Apache Druid](#)

Вместе с проектом предоставляются manifest файлы установки Apache Druid в среде Kubernetes.