ЮНИВЕРС
ДАТА

Руководство по установке

Юниверс SmartETL 1.x

2024

ООО «ЮНИВЕРС ДАТА» оставляет за собой право вносить изменения в настоящий документ без предварительного уведомления.

Данный документ и его отдельные части в любом порядке их расположения не подлежат воспроизведению, публикации и передаче третьим лицам (вне зависимости от конечной цели совершения указанных действий) без письменного разрешения ООО «ЮНИВЕРС ДАТА».

Редакция от 02.12.2024.

Содержание

Аннотация

Документ предназначен для получения сведений, необходимых для установки Юниверс SmartETL.

В документе описан порядок действий по установке и настройке компонентов системы.

Документ рассчитан на системных администраторов.

1 Системные требования и состав дистрибутива

1.1 Требования

Системные требования для SmartETL зависят от объема данных, которые требуется обработать. Для определения системных требований используйте рекомендации по масштабированию, указанные по ссылке: https://community.cloudera.com/t5/Community-Articles/NiFi-Sizing-Guide-Deployment-Best-Practices/ta-p/246781.

Пример конфигурации для 18 млн. записей (~10 кб каждая):

- Процессор: 16 virtual cores.
- Оперативная память: 32 gb.
- Хранение памяти: SSD.

Таблица 1 – Программные требования

Сервер	*Клиент*
Должна быть установлена одна из следующих операционных систем: • Ред ОС 8; • Ubuntu Server 20.04 LTS; • ALT Server 10.1; • Astra Linux Special Edition (релиз "Смоленск")	Клиент может быть запущен на: • Windows не ниже версии 8.1; • ОС семейства Linux с графическим интерфейсом и поддержкой указанных версий интернет-браузеров; • Mac OS X Необходимо использовать любой из интернет-браузеров: • Google Chrome (v. 99+): рекомендовано; • Microsoft Edge (v. 99+); • Mozilla Firefox (v. 99+)

Технологический стек решения

Зависимости для Подсистемы трансформации данных:

- Веб-сервер (например, nginx);
- Node.js 14.0.0.
- Npm 6.0.0.
- NiFi 1.15.3
- NiFi Registry 1.15.3
- Java JDK 11.

- PostgreSQL 14.1: БД модулей.

Зависимости для Подсистемы обеспечения служебных функций:

- Keycloak Server 16.1.1: управление доступом.
- Prometheus 2.33.4: мониторинг.
- PostgreSQL 14.1: БД модулей.
- Elasticsearch 7.14.2.
- Kibana 7.14.2.

Зависимости для Подсистемы хранения данных:

- Minio RELEASE.2021-10-23T03-28-24Z.hotfix.f7ed7b98e: хранилище версий данных.
- PostgreSQL 14.1: БД метаданных для хранилища версий данных.
- RabbitMQ 3.10.1: очередь сообщений.

1.2 Дистрибутив

smartetl.zip – Основной архив поставки

- **smartetl-model-registry-XXXXXXXX.zip** – Архив реестра моделей (backend)
  - **smartetl-model-registry** – Содержимое архива реестра моделей (backend)
- **smartetl-model-registry-ui-XXXXXXXX.zip** – Архив реестра моделей (frontend)
  - **smartetl-model-registry-ui** – Содержимое архива реестра моделей (frontend)
- **smartetl-nifi-XXXXXXXX.zip** – Архив расширений Apache NiFi
  - **smartetl-nifi** – Содержимое архива расширений Apache NiFi
    - **lib** – Библиотеки для Apache NiFi
  - **smartEtlExampleTemplate.xml** – Шаблон процессоров Apache NiFi

---

В состав дистрибутива не входят пакеты для установки окружения системы.
XXXXXXXX – хэш-код сборки.

---

2 Подготовка к установке

2.1 Варианты установки

Юниверс SmartETL может быть установлена двумя способами:

- Путем развертывания Docker-образа и его настройки (все компоненты будут установлены на один сервер. Подходит для ознакомления с системой или настройки системы). Инструкцию см. в разделе 3.
- Через ручную установку каждого из компонентов системы (компоненты могут быть установлены как на один сервер, так и на несколько разных). Инструкцию см. в разделе 4.

2.2 Порядок действий

Установка компонентов Юниверс SmartETL возможна в произвольном порядке, однако, рекомендуется придерживаться последовательности действий, обозначенных разработчиком системы.

Последовательность выглядит следующим образом:

1. Подсистема трансформации данных.
    - Установка Модуля реестра моделей.
    - Установка Модуля реестра Apache NiFi.
    - Установка Модуля трансформации данных.
    - Импорт шаблона SmartETL в Модуль трансформации данных.
2. Подсистема хранения данных.
    - Установка Хранилища версий данных.
    - Установка БД метаданных для версий данных.
    - Установка Очереди данных.
    - Установка Модуля работы с версиями данных.
3. Подсистема обеспечения служебных функций.
    - Установка БД модуля управления доступом.
    - Установка Модуля управления доступом.
    - Установка Модуля сбора и отображения логов.
4. Установка систем, с которыми планируется интеграция. Например, Юниверс MDM или Юниверс DQ.
5. Настройка взаимодействия между подсистемами.

2.3 Выбор архитектуры решения

При выборе конкретной реализации архитектуры следует опираться:

- На количество доступных вычислительных мощностей;
- На объем данных, которые планируется обрабатывать;
- На цели и задачи внедрения SmartETL.

Указанные факторы могут повлиять, например, на схему установки PostgreSQL. Возможные схемы:

- 1 инстанс PostgreSQL = 1 сервис. Рекомендуемая схема.
- 1 инстанс PostgreSQL на все сервисы SmartETL. При размещении всех БД на одном сервере следует использовать сервер-реплику (сервера должны размещаться на разных физических машинах).

3 Установка через Docker

3.1 Процедура установки

Установка выполняется из каталога *.setup*. Docker-образ настроен для работы на одном сервере.

Предварительно на сервер должны быть установлены:

- Docker;
- Docker-compose;
- Python3;
- Java11;
- Jq.

Для запуска установки:

1. Создайте ключи, если их нет.

```
cd .setup/cert/
./certs.sh <server-dns-name> <nifi.version>
```

2. Если для Apache Nifi меняется hostname, то необходимо это изменение сделать и в файле certs.sh.
3. В результате шага 1 будут созданы ключи для установки в каталоге *.setup/cert/data*.
4. Скопируйте файл *init.yml.sample* в *init.yml* и измените настройки на актуальные.
5. Запустите скрипт:

```
cd .setup
sudo ./init.sh
```

6. Перейдите в каталог, где настроено рабочее окружение (по умолчанию */opt/smartetl*).
7. Проверьте, какие из контейнеров приложения запущены. Список доступных сервисов будет выведен в конце установки. Команда:

```
docker-compose ps
```

Пример:

```
текущие настройки:
   default user:
      name:      user
      password:  password
   keycloak:
```

```
      url:       https://<WEB_DOMAIN_NAME>/keycloak/auth
      user:      user
      password:  password
   nifi:
      url:       https://<WEB_DOMAIN_NAME>/nifi
   nifi-registry:
      url:       https://<WEB_DOMAIN_NAME>/nifi-registry
   model-registry:
      url:       http://<WEB_DOMAIN_NAME>:3001
   minio:
      url:       http://<WEB_DOMAIN_NAME>:9000
   kibana:
      url:       https://<WEB_DOMAIN_NAME>/kibana
   prometheus:
      url:       https://<WEB_DOMAIN_NAME>/prometheus
   rabbitmq ui:
      url:       https://<WEB_DOMAIN_NAME>/rabbitmq
каталог для данных nifi:
      на хосте:  /opt/smartetl/nifi/volumes/data/import
      внутри контейнера (внутри nifi): /opt/data/import
```

8. После развертывания компоненты Юниверс SmartETL могут быть запущены и готовы для первичной настройки.

Известные проблемы

В случае, если не удается войти в Minio, перезапустите контейнер:

```
docker-compose stop minio
docker-compose up -d
```

3.2 Настройка docker-сборки

После развертывания образа необходимо настроить компоненты системы.

3.2.1 Настройки Apache NiFi

1. Запустите сервис Apache NiFi. Адрес строится следующим образом: https://SERVER_HOSTNAME/nifi/.

2. При загрузке автоматически произойдет переадресация на сервис авторизации Keycloak. Используйте логин и пароль admin / nifi для входа. Вход в сам Keycloak осуществляется по паролю admin / admin.
3. Нажмите кнопку ☰ в правом верхнем углу экрана.
4. Выберите пункт Users.
5. В модальном окне создайте необходимых пользователей и назначьте им группы.
6. После завершения закройте модальное окно.
7. Снова нажмите кнопку ☰ в правом верхнем углу экрана.
8. Выберите пункт Policies.
9. Добавьте политики для пользователей.
10. Импортируйте основной шаблон Apache NiFi. Для этого выполните действия:
    - Выберите действие Upload template: либо через область Operate, расположенную в левой части экрана; либо через контекстное меню, вызываемое ПКМ в произвольном месте рабочей области.
    - В результате откроется модальное окно загрузки.
    - Нажмите кнопку поиска в строке Select template.
    - При помощи проводника выберите шаблон в формате .xml (пример имени: smartEtlExampleTemplate).
    - В модальном окне нажмите кнопку Upload.
    - При успешной загрузке будет отображено соответствующее сообщение.
    - В результате шаблон будет доступен в системе для создания на его основе процессоров.
11. Создайте новый процессор из готового шаблона. Для этого выполните действия:
    - Перейдите в требуемую группу рабочей области (если необходимо).
    - Наведите курсор на иконку добавления шаблонов (Рисунок 1), расположенную в левом верхнем углу экрана.
    - Зажмите ЛКМ и перетащите иконку на рабочую область.
    - В результате действия откроется список добавления шаблонов.
    - Выберите шаблон из списка.
    - Нажмите кнопку Add.
    - В результате будут добавлены блоки и настройки, записанные в шаблон (как правило, собранные в отдельную группу). В любой группе допустимо использовать несколько одинаковых групп, созданных из шаблона.

- После добавления содержимое может быть отредактировано. Это не повлияет на шаблон, из которого содержимое создано.

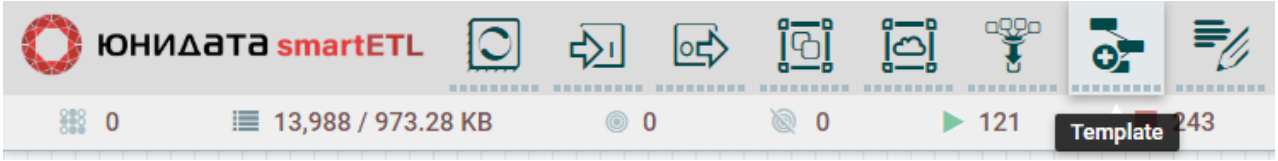

Рисунок 1 – Иконка добавления шаблонов

12. Настройте переменные для созданного процессора. Для этого выполните действия:
    - Наведите курсор на процессор (на корневую группу процессора) и нажмите ПКМ.
    - В контекстном меню выберите Variables.
    - Замените ссылки на актуальные для переменных: dataManagerURL, keycloakURL, modelRegistryURL, rabbitmqUrl, rabbitmqPort, UniverseDQWSDL, UniverseURL, elasticsearchURL, minioUrl, nifiSystemLogsPath.
    - Войдите в группу процессора. Дальнейшие действия будут производиться внутри этой группы.
    - В группе процессоров Read Json Data from Files откройте Variables и замените путь в переменной dataDirectory.
    - В группе процессоров Read Excel Data from Files откройте Variables и замените путь в переменной dataDirectory.
    - В группе процессоров Data Model From XLSX File -> Read One Excel File откройте Variables и замените путь в переменной dataDirectory.
    - Задайте пароль процессору UniverseClientService.
    - Задайте client secret процессору ModelRegistryService.
    - Задайте пароль к RabbitMQ в процессоре PublishAMQP.
    - Задайте пароль к RabbitMQ в процессоре ConsumeAMQP.
    - Задайте пароль к minio в процессоре FetchS3Object в репроцессинге.
    - Нажмите ПКМ в произвольном месте рабочей области.
    - В контекстном меню выберите пункт Configure.
    - Перейдите во вкладку Controller Services и убедитесь, что все процессоры имеют статус Enable.

3.2.2 Настройки очереди сообщений

1. Войдите в сервис RabbitMQ. Адрес строится следующим образом: **https://<WEB_DOMAIN_NAME>/rabbitmq**.

2. Создайте новую постоянную очередь сообщений для потока данных Apache Nifi (Data Flow). По умолчанию Routing Key имеет значение - smart-etl-output-route.

3.2.3 Настройка в файловой системе сервера

1. Создайте символическую ссылку на каталог /opt/smartetl/nifi/volumes/data/import. Адрес ссылки может быть любым.

3.2.4 Настройки Elasticsearch

1. Войдите в Elasticsearch. Для этого в сервисе Apache NiFi кликните по логотипу Kibana, расположенному в правом верхнем углу экрана (Рисунок 2).

2. Нажмите кнопку ☰ в левом верхнем углу экрана.

3. В выпадающем меню выберите пункт Management > Dev Tools.

4. Создайте индекс nifi-data-provenance (содержимое индекса см. ниже). Для этого вставьте содержимое индекса и нажмите кнопку ▷ *Click to sent request* (в правом верхнем углу окна ввода запроса). В результате действия в консоли отобразится результат запроса и статус 200.

5. Создайте индекс nifi-app-logs (содержимое индекса см. ниже). Для этого вставьте содержимое индекса и нажмите кнопку ▷ *Click to sent request*. В результате действия в консоли отобразится результат запроса и статус 200.

6. Нажмите кнопку ☰ в левом верхнем углу экрана.

7. В выпадающем меню выберите пункт Stack Management > Saved objects.

8. Нажмите кнопку Import, расположенную в правом верхнем углу экрана.

9. Выберите в файловой системе шаблон для дашбордов и загрузите его. Шаблон находится в комплекте поставки.

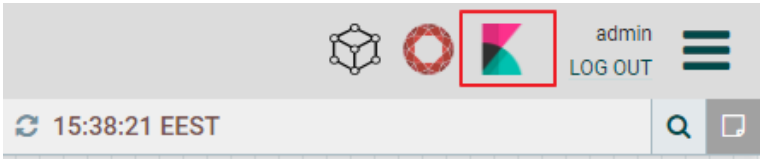

Рисунок 2 – Кнопка перехода в Elasticsearch / Kibana.

Индекс nifi-data-provenance

```
PUT /nifi-data-provenance
{
  "settings": {
    "number_of_shards": 2,
    "number_of_replicas": 1
  },
  "mappings": {
    "properties": {
        "actorHostname": {
          "type": "keyword"
        },
        "application": {
          "type": "keyword"
        },
        "batchId": {
          "type": "keyword"
        },
        "childIds": {
          "type": "keyword"
        },
        "componentId": {
          "type": "keyword"
        },
        "componentName": {
          "type": "keyword"
        },
        "componentType": {
          "type": "keyword"
        },
        "contentURI": {
          "type": "keyword"
```

```
},
"details": {
  "type": "keyword"
},
"durationMillis": {
  "type": "long"
},
"entityId": {
  "type": "keyword"
},
"entitySize": {
  "type": "long"
},
"entityType": {
  "type": "keyword"
},
"eventId": {
  "type": "keyword"
},
"eventOrdinal": {
  "type": "long"
},
"eventType": {
  "type": "keyword"
},
"lineageStart": {
  "type": "date"
},
"parentIds": {
  "type": "keyword"
},
"platform": {
  "type": "keyword"
},
"previousAttributes": {
```

```
      "type": "nested",
      "properties": {
        "name": {
          "type": "keyword"
        },
        "value": {
          "type": "keyword"
        }
      }
    },
    "previousContentURI": {
      "type": "keyword"
    },
    "previousEntitySize": {
      "type": "long"
    },
    "processGroupId": {
      "type": "keyword"
    },
    "processGroupName": {
      "type": "keyword"
    },
    "runId": {
      "type": "long"
    },
    "timestamp": {
      "type": "date"
    },
    "timestampMillis": {
      "type": "date"
    },
    "transitUri": {
      "type": "keyword"
    },
    "updatedAttributes": {
```

```
            "type": "nested",
            "properties": {
              "name": {
                "type": "keyword"
              },
              "value": {
                "type": "keyword"
              }
            }
          },
          "uuid": {
            "type": "keyword"
          }
        }
    }
}
```

Индекс nifi-app-logs

```
PUT /nifi-app-logs
{
  "settings": {
    "number_of_shards": 2,
    "number_of_replicas": 1
  },
  "mappings": {
    "properties": {
        "_raw": {
          "type": "text"
        },
        "date": {
          "type": "date"
        },
        "level": {
          "type": "keyword"
```

```
        },
        "logger": {
          "type": "keyword"
        },
        "message": {
          "type": "text"
        },
        "thread": {
          "type": "keyword"
        },
        "timestamp": {
          "type": "date"
        }
      }
  }
}
```

3.2.5 Настройки Data provenance

1. Войдите в сервис Apache NiFi, если это не сделано ранее.
2. Создайте пользователя с именем "CN=nifi, OU=NIFI".
3. Настройте следующие policy для пользователя "CN=nifi, OU=NIFI":
   - receive data via site-to-site для порта DP event
   - retrieve site-to-site details в основном меню ☰.
   - query provenance в основном меню ☰.
4. Создайте Reporting task SiteToSiteProvenanceReportingTask с набором настроек. Пример настроек на скриншотах ниже (Рисунок 3 – 5).
   - Значения URL должны быть актуальны для вашей системы.
   - Параметр Communications Timeout зависит от ресурсов сервера, подбирается опытным путём. Чем меньше ресурсов, тем больше может быть таймаут.
5. Пароли для сертификатов можно получить через через команды:

```
sudo docker ps (найти id контейнера nifi)
sudo docker exec -ti -u root <id контейнера nifi> bash
```

```
cat /opt/nifi/nifi-current/conf/nifi.properties
```

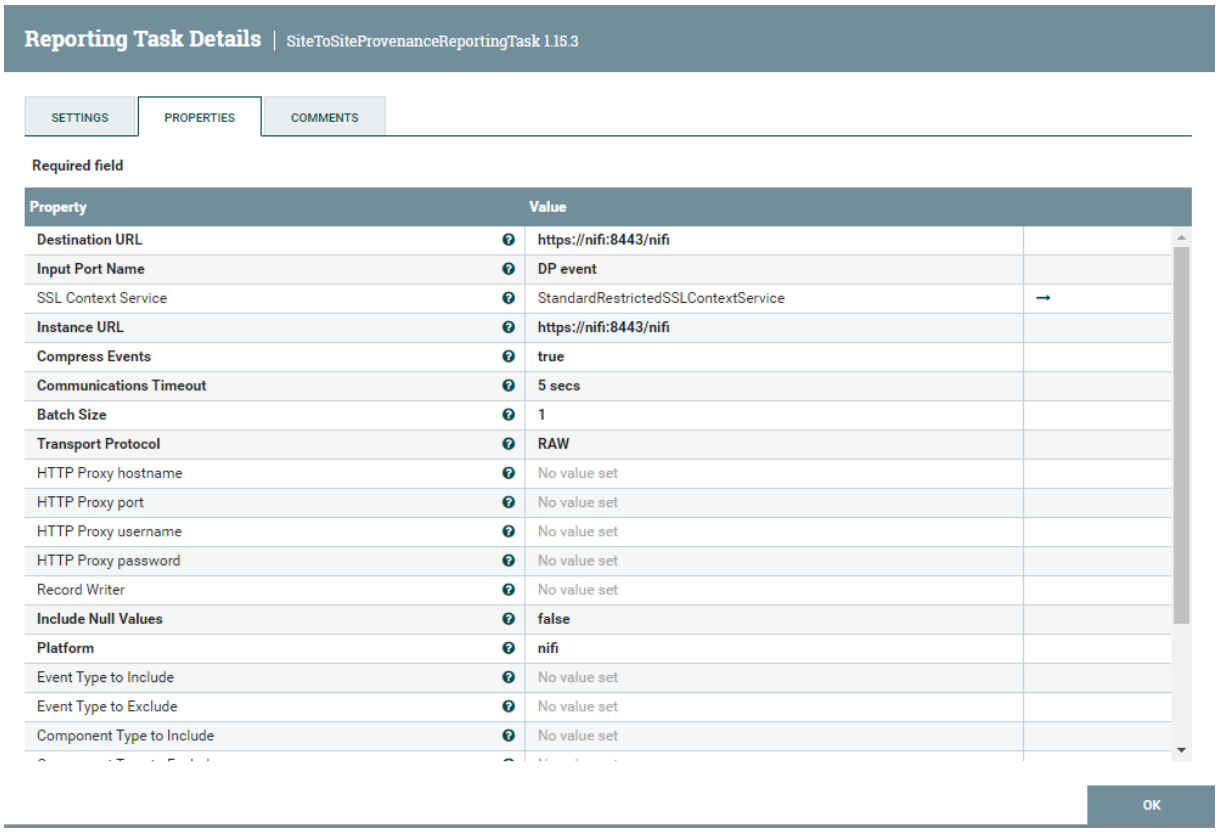

Рисунок 3 – Настройки SiteToSiteProvenanceReportingTask, часть 1

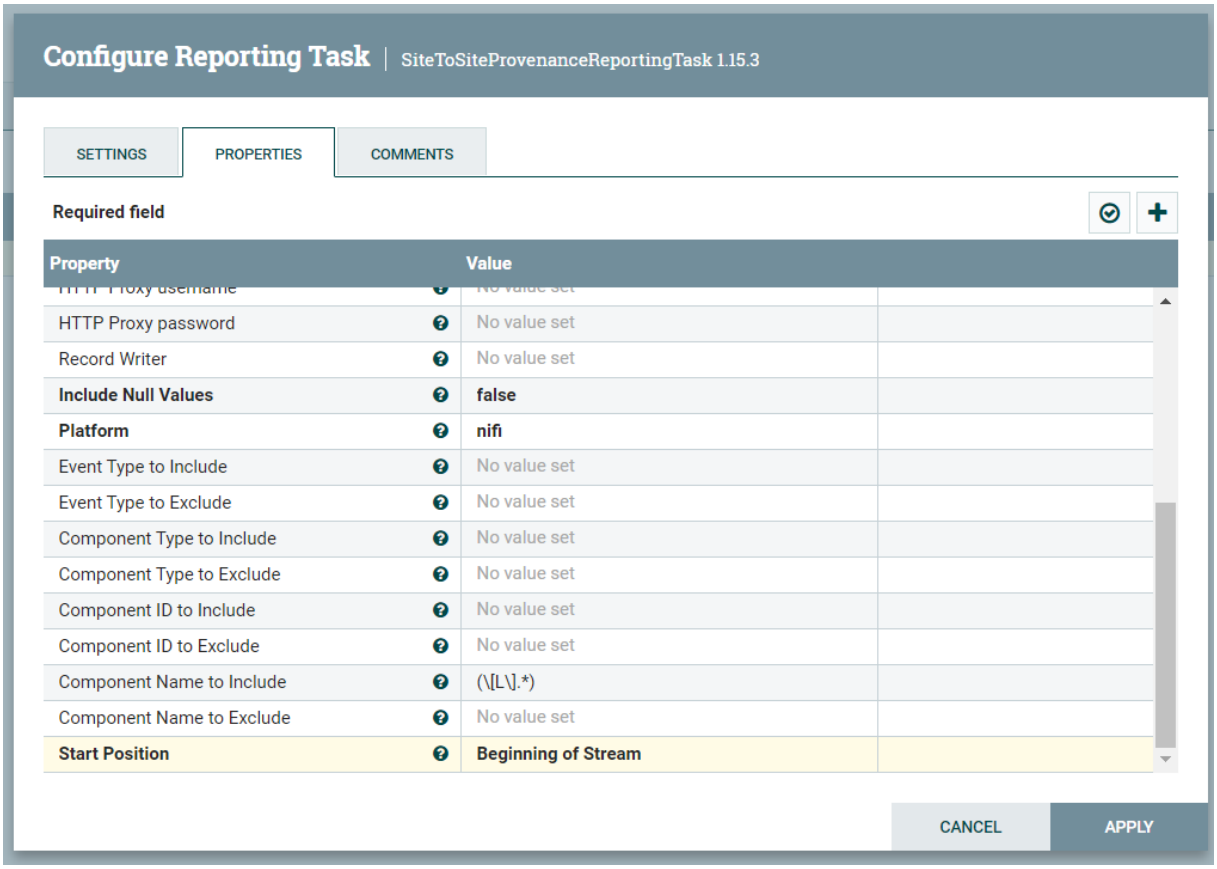

Рисунок 4 – Настройки SiteToSiteProvenanceReportingTask, часть 2

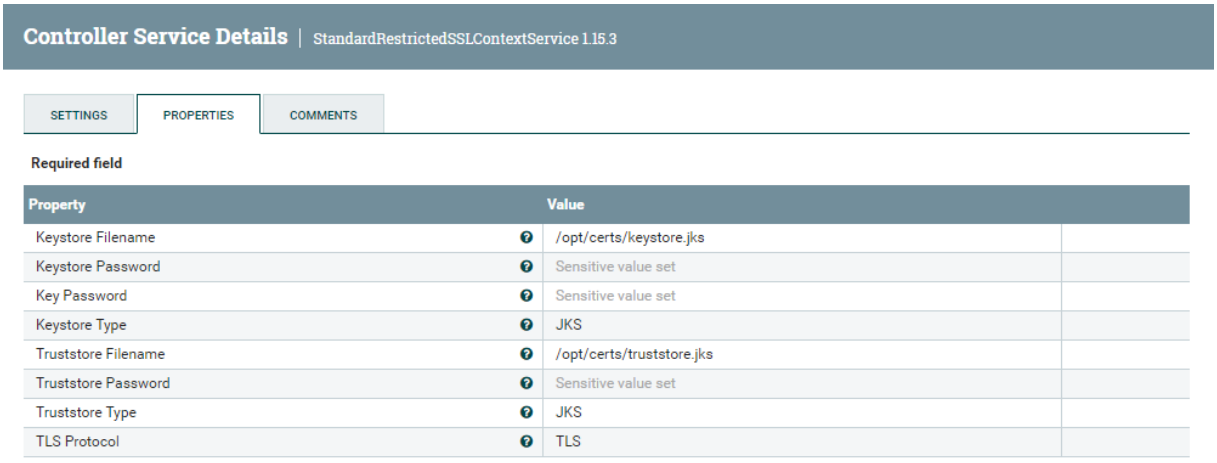

Рисунок 5 – Настройки SiteToSiteProvenanceReportingTask, часть 3

4 Установка вручную

4.1 Подсистема трансформации данных

4.1.1 Установка Модуля реестра моделей

В инструкции указан пример установки для Unix-подобных ОС на примере CentOS версии 7 и выше.

Backend

Этап 1. Подготовка сервера

1. Установите PostgreSQL 14.1

```
yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y
```

```
yum -y install postgresql14-server postgresql14-contrib/usr/pgsql-14/bin/postgresql-14-setup initdb
```

2. Проинициализируйте БД. Инициализация БД будет использовать установленную локаль.

```
/usr/pgsql-13/bin/postgresql-14-setup initdb
```

3. Подключитесь к серверу с использованием учётной записи суперпользователя (root):

```
ssh root@<Имя или IP адрес сервера>
```

4. Перейдите в учётную запись пользователя postgres:

```
su root
```

```
su postgres
```

5. Войдите в PostgreSQL через учетную запись:

```
psql -U postgres
```

6. Создайте БД:

```
create database smartetl_model_registry;
```

7. После создания БД PostgreSQL может быть перезапущен.

8. Убедитесь, что установлены следующие зависимости. При необходимости, установите их:

- Node.js версии 14.0.0 и выше.
- Npm версии 6.0.0 и выше.

Этап 2. Установка дистрибутива

9. Скопируйте содержимое дистрибутива SmartETL в требуемый каталог на сервере. Например, `/opt/smartetl-model-registry`.

10. Убедитесь, что у текущего пользователя есть права на каталог (минимально права на чтение).

11. Пропишите данные подключения к БД и прочие настройки конфигурации в .env-файл. В том числе укажите в ALLOW_ORIGINS полные url-адреса клиентского приложения и Apache NiFi для разрешения кросс-доменных запросов. Например, `ALLOW_ORIGINS="http://smartetl.local:3001;https://localhost:8443/.`

12. Зайдите в каталог с содержимым дистрибутива.

13. Установите зависимости npm:

```
npm i
```

14. Выполните скрипт для установки зависимостей npm:

```
npm run seed
```

15. Выполните скрипт для инициализации данных в БД:

```
npm migration:run
```

16. Опционально. Глобально установите менеджер процессов PM2 для nodejs. Менеджер необходим для удобного запуска и остановки процесса nodejs. Инструкция: https://pm2.keymetrics.io/docs/usage/quick-start/

```
npm i pm2 -g
```

17. Запустите реестр моделей. Обычный старт:

```
node src/main.js
```

Старт через PM2:

```
pm2 start src/main.js
```

Frontend

Для установки Frontend-клиента:

1. Установите и настройте любой веб-сервер. Пример конфигурации для nginx:

```
server {
  listen 3001; //пример порта
  root /var/www/html/smartetl-model-registry-ui;
  index index.html index.htm;
    location / {
```

```
        try_files $uri $uri/ /index.html;
    }
}
```

2. Поместите содержимое приложения на сервер. В указанном примере приложение находится в каталоге /var/www/html/smartetl-model-registry-ui.

3. Замените URL сервера для приложения. Для этого в каталоге приложения откройте файл customer.json и отредактируйте параметр:

```
"serverUrl": "<url сервера>"
```

4. Добавьте полный URL клиентского сервера в .env файл сервера приложения (backend) для разрешения кросс-доменных запросов, если это не сделано ранее:

```
ALLOW_ORIGINS="http://smartetl.local:3001"
```

где 3001 – номер порта, указанный в конфигурации веб-сервера. Если используется порт 80, то в ALLOW_ORIGINS порт не указывается.

5. Проверьте корректность установки реестра моделей, пройдя по URL. Если открывается страница приложения, то все установлено верно.

4.1.2 Установка Модуля реестра Apache NiFi

Этап 1. Установка NiFi Registry

Выполните действия:

1. Установите Java

```
yum install java-1.11.0-openjdk -y
```

2. Скачайте дистрибутив NiFi из архива поставки.

```
yum install wget -y
```

```
cd /data/app/ && wget https://archive.apache.org/dist/nifi/1.15.3/nifi-registry-1.15.3-bin.tar.gz
```

3. Распакуйте архив после скачивания.

```
tar -xvf nifi-registry-1.15.3-bin.tar.gz
```

4. Укажите JAVA_HOME. Для этого отредактируйте файл `vi /data/app/nifi-registry-1.15.3/bin/nifi-env.sh`Параметры файла:

```
export JAVA_HOME=/usr/lib/jvm/jre-11-openjdk
```

5. Если используется firewall, то откройте порт, указанный в nifi.properties.

```
firewall-cmd --add-port=18080/tcp
```

```
firewall-cmd --permanent --add-port=18080/tcp
```

6. Добавьте приложение как сервис:

```
/data/app/nifi-registry-1.15.3/bin/nifi-registry.sh install
```

Этап 2. Установка БД и NiFi Registry для работы с БД

Выполните действия:

7. Установите PostgreSQL 14.1

```
yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y
```

```
yum -y install postgresql14-server postgresql14-contrib/usr/pgsql-14/bin/postgresql-14-setup initdb
```

8. Проинициализируйте БД. Инициализация БД будет использовать установленную локаль.

```
/usr/pgsql-13/bin/postgresql-14-setup initdb
```

9. Подключитесь к серверу с использованием учётной записи суперпользователя (root):

```
ssh root@<Имя или IP адрес сервера>
```

10. Перейдите в учётную запись пользователя postgres:

```
su root
```

```
su postgres
```

11. Войдите в PostgreSQL через учетную запись:

```
psql -U postgres
```

12. Создайте БД:

```
create database smartetl_nifi_registry;
```

13. После создания БД PostgreSQL может быть перезапущен.

14. Обновите содержимое файла `/var/lib/pgsql/14/data/postgresql.conf` на:

```
listen_addresses = '*'
```

15. В случае, если БД выделена только под NiFi Registry, отредактируйте файл `/var/lib/pgsql/9.6/data/pg_hba.conf`:

```
local smartetl_nifi_registry smartetl_nifi_registry scram-sha-256
```

```
local all all ident
```

```
host all all 127.0.0.1/32 scram-sha-256
```

16. Запустите PostgreSQL:

```
systemctl enable postgresql-14
```

```
systemctl start postgresql-14
```

17. Создайте пользователя smartetl_nifi с требуемым паролем:

```
sudo -u postgres psql -c 'create database smartetl_nifi_registry;'
```

```
sudo -u postgres psql -c "CREATE USER smartetl_nifi_registry WITH
PASSWORD 'smartetl_nifi_registry'; GRANT ALL PRIVILEGES ON DATABASE
smartetl_nifi_registry to smartetl_nifi_registry;"
```

18. Подложите драйвер для БД postgresql-42.3.2.jar в каталог `/data/app/nifi-registry-1.15.3/`:

```
wget https://jdbc.postgresql.org/download/postgresql-42.3.2.jar
```

19. Отредактируйте файл `/data/app/nifi-registry-1.15.3/conf/nifi-registry.properties`. Замените следующие параметры:

```
nifi.registry.db.url=jdbc:postgresql://127.0.0.1/smartetl_nifi_registry
```

```
nifi.registry.db.driver.class=org.postgresql.Driver
```

```
nifi.registry.db.driver.directory=/data/app/nifi-registry-1.15.3/
```

```
nifi.registry.db.username=smartetl_nifi_registry
```

```
nifi.registry.db.password=smartetl_nifi_registry
```

20. Запустите NiFi Registry:

```
service nifi-registry start
```

```
# Проверка статуса
```

```
service nifi-registry status
```

21. Если необходимо добавить сервис в автозагрузку, то отредактируйте файл crontab:

```
sudo crontab -e
```

- Добавьте следующую строку в конце файла и сохраните файл:

```
@reboot service nifi-registry start
```

22. Для проверки web-порта используйте команду ниже. Сервис может начать отвечать спустя несколько минут после запуска.

```
curl -s http://127.0.0.1:18080/nifi-registry/| head
```

4.1.3 Установка Модуля трансформации данных Apache NiFi

Этап 1. Распаковка и настройка

Выполните действия:

1. Установите Java JDK:

```
yum install java-1.11.0-openjdk -y
```

2. Скопируйте дистрибутив NiFi из архива поставки:

```
yum install wget -y
cd /opt/ && wget https://archive.apache.org/dist/nifi/1.15.3/nifi-1.15.3-bin.tar.gz
```

3. Распакуйте архив:

```
tar -xvf nifi-1.15.3-bin.tar.gz
```

4. Скопируйте файлы из дистрибутива /NiFi/conf в /opt/nifi-1.15.3/conf/

5. Измените параметры в файле bootstrap.conf, выставив потребление RAM (Xms/Xmx) в зависимости от характеристик сервера. Рекомендуется использовать не более 60% от общей памяти. Пример команды:

```
vi /opt/nifi-1.15.3/conf/bootstrap.conf
```

Пример значений параметров:

```
# JVM memory settings
java.arg.2=-Xms4G
java.arg.3=-Xmx20G
```

6. В конец файла bootstrap.conf добавьте параметр:

```
# Fixes encoding for russian symbols
java.arg.57=-Dfile.encoding=UTF-8
```

7. Для замены порта и доступных интерфейсов для web доступа отредактируйте файл nifi.properties:

```
vi /opt/nifi-1.15.3/conf/nifi.properties
```

Параметры файла:

```
### доступные интерфейсы, 0.0.0.0 - для всех
nifi.web.https.host=0.0.0.0
nifi.web.https.port=8443
```

8. Укажите JAVA_HOME. Для этого отредактируйте файл `/data/app/nifi-1.15.3/bin/nifi-env.sh`. Параметры файла:

```
export JAVA_HOME=/usr/lib/jvm/jre-11-openjdk
```

9. Замените порты и доступные интерфейсы. Для этого в файле `/data/app/nifi-1.15.3/conf/nifi.properties` внесите изменения в следующих параметрах:

```
# Site to Site properties
nifi.remote.input.host=0.0.0.0
nifi.remote.input.secure=false
nifi.remote.input.socket.port=1000
nifi.remote.input.http.enabled=true
```

```
### доступные интерфейсы, 0.0.0.0 - для всех

nifi.web.http.host=0.0.0.0
nifi.web.http.port=8081

nifi.web.https.host=
nifi.web.https.port=

nifi.security.keystore=
nifi.security.keystoreType=
nifi.security.keystorePasswd=
nifi.security.keyPasswd=
nifi.security.truststore=
nifi.security.truststoreType=
nifi.security.truststorePasswd=

nifi.security.ocsp.responder.url=
nifi.security.ocsp.responder.certificate=
```

10. Если используется firewall, то откройте порт, указанный в nifi.properties.

```
firewall-cmd --add-port=8443/tcp
firewall-cmd --permanent --add-port=8443/tcp
```

11. Задайте пользователя и пароль для web-интерфейса и добавьте приложение как сервис:

```
/data/app/nifi-1.15.3/bin/nifi.sh set-single-user-credentials nifi_user
nif_passwrd__
```

Этап 2. Настройка ОС

Выполните действия:

1. Настройте ОС, используя рекомендации Apache NiFi: https://nifi.apache.org/quickstart.html
2. Скорректируйте файл /etc/security/limits.conf, добавив или изменив следующие параметры:

```
* hard nofile 50000
```

```
* soft nofile 50000
* hard nproc 10000
* soft nproc 10000
```

3. Выполните команду:

```
sudo sysctl -w net.ipv4.ip_local_port_range="10000 65000"
```

4. Скорректируйте файл/etc/sysctl.conf, добавив или изменив следующий параметр:

```
vm.swappiness = 0
```

Этап 3. Кластерная конфигурация NiFi

Актуально при использовании Apache NiFi на нескольких узлах.

1. Отредактируйте файл nifi.properties:

```
vi /opt/nifi-1.15.3/conf/nifi.properties
```

2. Измените параметры в файле. Вместо nifi_current_host укажите hostname текущего сервера:

```
nifi.cluster.is.node=true
nifi.cluster.node.address=nifi_current_host
nifi.cluster.node.protocol.port=12000
nifi.remote.input.host=nifi_current_host
nifi.cluster.protocol.is.secure=true
```

Этап 4. Подготовка к запуску

Выполните действия:

1. Скопируйте .nar файлы из дистрибутива /NiFi/lib/ в каталог nifi opt/nifi-1.15.3/lib/
2. Запустите NiFi.

```
service nifi start
```

3. Проверьте статус сервиса.

```
service nifi status
```

4. Для добавления приложения в автозагрузку:
   - Отредактируйте файл cron с помощью команды:

```
sudo crontab -e
```

   - Добавьте следующую строку в конец файла и сохраните файл:

```
@reboot service nifi start
```

- Для просмотра crontab используйте команду:

```
sudo crontab -l
```

5. Для проверки web-порта используйте команду ниже. Сервис может начать отвечать спустя несколько минут после запуска.

```
curl https://$(uname -n):8443/nifi/ -s --insecure | head
```

6. Откройте NiFi в браузере. Адрес строится следующим образом:

```
https://SERVER_HOSTNAME:8443/nifi/
```

Настройка конфигурации NiFi

Выполните действия:

1. Операция выполняется для каждого узла NiFi.
2. В интерфейсе Apache NiFi откройте настройки. Для этого кликните по кнопке ☰ в правом верхнем углу экрана, выберите пункт Controller Settings, вкладка General.
    - Укажите Maximum Timer Driven Thread Count = (кол-во виртуальных ядер сервера) * 2.
    - Укажите Maximum Event Driven Thread Count = 5.
    - Подключите NiFi Registry.
    - На вкладке Registry Clients добавьте url сервера с NiFi Registry. Пример: http://SERVER_HOSTNAME:18080.
    - Нажмите Apply.

4.1.4 Импорт шаблона в Apache NiFi

Шаблоны Apache NiFi используются для хранения информации о потоках NiFi, для их использования на нескольких узлах, другими разработчиками и т.д. SmartETL работает со специальным шаблоном, который должен быть импортирован в систему после установки всех компонентов.

Предварительные условия:

- Убедитесь, что установлен Реестр моделей.
- Убедитесь, что установлен Apache NiFi.
- Из комплекта поставки скопируйте файл шаблона в произвольный каталог. Шаблон может иметь имя вида smartEtlExampleTemplate.xml.

Чтобы **импортировать шаблон**:

1. Откройте Apache NiFi в браузере. Адрес строится следующим образом:

```
https://SERVER_HOSTNAME:8443/nifi/
```

2. Выберите действие Upload template: либо через область Operate, расположенную в левой части экрана; либо через контекстное меню, вызываемое ПКМ в произвольном месте рабочей области.
3. В результате откроется модальное окно загрузки.
4. Нажмите кнопку поиска в строке Select template.
5. При помощи проводника выберите шаблон в формате .xml (пример имени: smartEtlExampleTemplate). Шаблон должен быть заранее скопирован из комплекта поставки.
6. В модальном окне нажмите кнопку Upload.
7. При успешной загрузке будет отображено соответствующее сообщение.

Чтобы **добавить шаблон**:

1. На панели инструментов, расположенную в верхней части экрана, найдите иконку  Template.
2. Перетащите иконку на рабочую область.
3. В результате откроется модальное окно.
4. Выберите из выпадающего списка требуемый шаблон.
5. Нажмите кнопку Add.
6. В результате в рабочей области появится содержимое шаблона. Далее необходимо настроить процессор.
7. Настройте переменные для созданного процессора. Для этого выполните действия:
    - Наведите курсор на процессор (на корневую группу процессора) и нажмите ПКМ.
    - В контекстном меню выберите Variables.
    - Замените ссылки на актуальные для переменных: dataManagerURL, keycloakURL, modelRegistryURL, rabbitmqUrl, rabbitmqPort, UniverseDQWSDL, UniverseURL, elasticsearchURL, minioUrl, nifiSystemLogsPath.
    - Войдите в группу процессора. Дальнейшие действия будут производиться внутри этой группы.

- В группе процессоров Read Json Data from Files откройте Variables и замените путь в переменной dataDirectory.
- В группе процессоров Read Excel Data from Files откройте Variables и замените путь в переменной dataDirectory.
- В группе процессоров Data Model From XLSX File -> Read One Excel File откройте Variables и замените путь в переменной dataDirectory.
- Задайте пароль процессору UniverseClientService.
- Задайте client secret процессору ModelRegistryService.
- Задайте пароль к RabbitMQ в процессоре PublishAMQP.
- Задайте пароль к RabbitMQ в процессоре ConsumeAMQP.
- Задайте пароль к minio в процессоре FetchS3Object в репроцессинге.
- Нажмите ПКМ в произвольном месте рабочей области.
- В контекстном меню выберите пункт Configure.
- Перейдите во вкладку Controller Services и убедитесь, что все процессоры имеют статус Enable.

4.2 Подсистема хранения данных

4.2.1 Установка БД метаданных для версий данных

Выполните действия:

1. Установите PostgreSQL 14.1:

```
yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y
```

```
yum -y install postgresql14-server postgresql14-contrib/usr/pgsql-14/bin/postgresql-14-setup initdb
```

2. Проинициализируйте БД. Инициализация БД будет использовать установленную локаль.

```
/usr/pgsql-13/bin/postgresql-14-setup initdb
```

3. Подключитесь к серверу с использованием учётной записи суперпользователя (root):

```
ssh root@<Имя или IP адрес сервера>
```

4. Перейдите в учётную запись пользователя postgres:

```
su root
```

```
su postgres
```

5. Войдите в PostgreSQL через учетную запись:

```
psql -U postgres
```

6. Создайте БД:

```
create database smartetl_data_manager;
```

7. После создания БД PostgreSQL может быть перезапущен.

4.3 Подсистема обеспечения служебных функций

4.3.1 Установка БД модуля управления доступом

Выполните действия:

- Установите PostgreSQL 14.1 по инструкции, указанной в пп. 4.2.1 (действия 1 – 5).
- Создайте БД:

```
create database smartetl_iam;
```

- После создания БД PostgreSQL может быть перезапущен.

4.3.2 Установка Модуля управления доступом

В инструкции указан пример установки для Unix-подобных ОС на примере Ubuntu 18.04.

Этап 1. Установка JDK

1. Убедитесь, что установлена Java и JDK версии 8. Если необходимо установить Java, то используйте команды:

```
yum install java-1.8.0-openjdk -y
```

2. Проверьте корректность установки JDK командой:

```
java -version
```

Этап 2. Распаковка Keycloak Server

3. Найдите актуальную сборку Keycloak на официальном сайте: https://www.keycloak.org/downloads.html и скопируйте ссылку на скачивание. Ниже представлены примеры команд для версии 6.1.1.

4. Скачайте дистрибутив:

```
cd /data/app/
```

```
sudo wget https://downloads.jboss.org/keycloak/6.1.1/keycloak-
6.1.1.tar.gz
```

5. Распакуйте архив и переименуйте каталог с дистрибутивом:

```
sudo tar -xvzf keycloak-6.1.1.tar.gz
sudo mv keycloak-6.1.1 /data/app/keycloak
```

Этап 3. Создание пользователя и группы для Keycloak

6. Не рекомендуется запускать Keycloak под root правами. Поэтому необходимо создание нового пользователя и выделение ему соответствующих прав.

7. Создайте новую группу и пользователя

```
sudo groupadd keycloak
sudo useradd -r -g keycloak -d /data/app/keycloak -s /sbin/nologin
keycloak
```

8. Назначьте права доступа на домашний каталог Keycloak:

```
cd /data/app/
sudo chown -R keycloak: keycloak
sudo chmod o+x /data/app/keycloak/bin/
```

Этап 4. Создание служебного файла SystemD

9. Создайте каталог конфигурации для Keycloak в каталоге /etc:

```
cd /etc/
sudo mkdir keycloak
```

10. Скопируйте из дистрибутива файл конфигурации и переименуйте его:

```
sudo cp /data/app/keycloak/docs/contrib/scripts/systemd/wildfly.conf
/etc/keycloak/keycloak.conf
```

11. Скопируйте скрипт запуска Keycloak (launch.sh):

```
sudo cp /data/app/keycloak/docs/contrib/scripts/systemd/launch.sh
/data/app/keycloak/bin/
```

12. Выдайте пользователю keycloak права доступа к скопированному скрипту:

```
sudo chown keycloak: /data/app/keycloak/bin/launch.sh
```

13. Исправьте путь установки Keycloak внутри скрипта. Откройте launch.sh в текстовом редакторе:

```
sudo nano /data/app/keycloak/bin/launch.sh
```

14. Найдите строку if ["x$WILDFLY_HOME" = "x"]; then WILDFLY_HOME="/data/app/keycloak".

15. В строке укажите новое значение пути: if ["x$WILDFLY_HOME" = "x"]; then WILDFLY_HOME="/data/app/keycloak".

16. Сохраните и закройте файл.

17. Скопируйте из дистрибутива файл определения сервиса (wildfly.service) и переименуйте его:

```
sudo cp /data/app/keycloak/docs/contrib/scripts/systemd/wildfly.service
/etc/systemd/system/keycloak.service
```

18. Откройте keycloak.service в текстовом редакторе:

```
sudo nano /etc/systemd/system/keycloak.service
```

19. Найдите в файле секции Unit, Service, Install. Отредактируйте секции следующим образом:

```
[Unit]
Description=The Keycloak Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=/etc/keycloak/keycloak.conf
User=keycloak
Group=keycloak
LimitNOFILE=102642
PIDFile=/var/run/keycloak/keycloak.pid
ExecStart=/data/app/keycloak/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG
$WILDFLY_BIND
StandardOutput=null

[Install]
WantedBy=multi-user.target
```

20. Сохраните и закройте файл.

21. Обновите конфигурацию systemd и добавьте Keycloak в автозагрузку:

```
sudo systemctl daemon-reload
sudo systemctl enable keycloak
```

Этап 5. Запуск сервиса

22. Запустите Keycloak

```
sudo systemctl start keycloak
```

23. Для проверки запуска сервиса используйте команду ниже. Если статус active, то Keycloak запущен успешно.

```
sudo systemctl status keycloak
```

24. При необходимости доступен просмотр логов:

```
sudo tail -f /data/app/keycloak/standalone/log/server.log
```

25. Откройте Keycloak в браузере. Пример ссылки: http://localhost:8080/auth/

Этап 6. Создание суперпользователя

Так как в Keycloak изначально нет пользователей, то может понадобиться создание пользователя-администратора системы. Есть два варианта сделать это.

Вариант 1:

- Откройте Keycloak в браузере с того же компьютера, где установлен сервис (пример ссылки http://localhost:8080/auth/). Укажите имя пользователя и пароль, подтвердите действие. Если Keycloak будет открыт с другого компьютера, то создание пользователя будет недоступно.

Вариант 2:

- Перейдите в /data/app/keycloak/bin/ и запустите скрипт создания пользователя:

```
sudo /data/app/keycloak/bin/add-user-keycloak.sh -r master -u <username>
-p <password>
```

- При первом запуске Keycloak будет созданный пользователь получит все доступные права.
- Перезапустите Keycloak:

```
sudo systemctl restart keycloak
```

4.3.3 Настройка SSO для Модуля реестра моделей

Настройка единой точки аутентификации производится в Модуле управления доступом (через приложение Keycloak).

Принцип работы:

- При входе в реестр моделей пользователь, не авторизованный в Keycloak, будет перенаправлен на страницу логина Keycloak.

- При верном вводе логина и пароля пользователь будет возращен в реестр моделей, получив 2 токена – access и refresh.
- Теперь все запросы пользователя на backend будут содержать в заголовке bearer-токен, который представляет собой JWT access-токен, по которому backend-часть будет в свою очередь проверять пользователя в Keycloak. По умолчанию задана стратегия, при которой backend-часть делает это один раз и кэширует токен до даты его истечения).
- Если access-токен истекает, клиент реестра моделей автоматически получает новый access-токен и будет обращаться на backend реестра моделей уже по нему. Backend в свою очередь сделает валидацию пользователя по уже новому, пришедшему с клиента токену.

Предварительные условия:

- Убедитесь, что установлен Модуль реестра моделей.
- Убедитесь, что установлен Модуль управления доступом.
- Войдите в Модуль управления доступом (Keycloak).

Чтобы **настроить SSO**:

1. Создайте новый клиент для frontend части Модуля реестра моделей. В поле Client ID указывается имя приложения. Например, model-registry-ui.
2. Перейдите к настройке frontend (вкладка Settings). Установите следующие параметры в указанные значения:
    - Enabled: on.
    - Client protocol: openid-connect.
    - Access Type: public. Такой тип доступа нужен для того, чтобы клиентская часть инициировала страницу логина пользователя.
    - Standard Flow Enabled: on.
    - Direct Access Grants Enabled: on.
    - Root URL: [url приложения].
    - Valid Redirect URIs: [url приложения]/*.
    - Web Origins: [список url приложения].
3. Создайте новый клиент для backend части Модуля реестра моделей. В поле Client ID указывается имя приложения. Например, model-registry.

4. Перейдите к настройке frontend (вкладка Settings). Установите следующие параметры в указанные значения:

- Enabled: on.
- Client protocol: openid-connect.
- Access Type: bearer-only. Такой тип доступа позволяет работать с bearer-токеном, пришедшим в заголовке Authorization. При этом серверная часть реестра моделей будет обращаться в keycloak только при первом получении bearer-токена для его валидации. Далее этот токен будет закэширован на сервере реестра моделей до тех пор, пока не закончится его время жизни.

5. Перейдите к настройке реквизитов вход (вкладка Credentials). Установите следующие параметры:

- Client Authenticator: Client Id and Secret.
- Secret: [содержание секрета]. Секрет будет использоваться backend-частью реестра моделей для идентификации приложения в Keycloak.

5 Проверка корректности установки

Юниверс SmartETL считается корректно установленной, если установлены все компоненты системы согласно архитектуре решения, принятой ко внедрению в ИТ-ландшафт заказчика. Перечень компонентов указан в пп. 2.2.

При ручной установке, в инструкции к установке каждого из компонентов указаны способы проверки работоспособности.

Проверка версий:

- Для Модуля реестра моделей: номер версии выводится в правом верхнем углу приложения.
- Для Модуля трансформации данных: кнопка меню > пункт About.

Список сокращений и условные обозначения

Принятые термины и определения

ETL	аббревиатура Extract, Transform, Load. Вид процессов управления данными, который состоит из этапов: извлечение, трансформация и выгрузка данных.
Авторизация	предоставление учетной записи прав на выполнение определенных действий в процессе входа в платформу.
Аутентификация	проверка подлинности логина/пароля или идентификатора учетной записи.
БД, база данных	совокупность данных, хранимых в соответствии со схемой данных.
ЛКМ	Левая кнопка мыши
ПКМ	Правая кнопка мыши

Условные обозначения

Важное уточнение

Примечание

Фрагменты программного кода, либо содержимого файла:

```
Universe.search.nodes.addresses=localhost:9300
Universe.search.cluster.name=elasticsearch-cluster
Universe.search.index.prefix = default
```