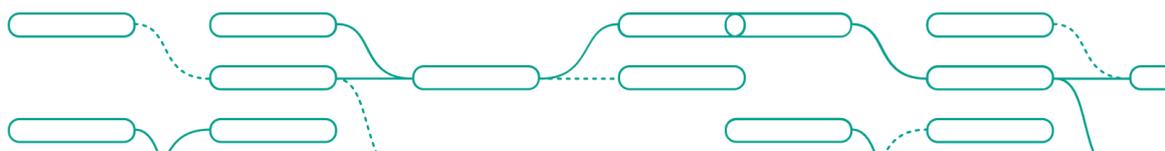


Arenadata Catalog

Инструкция по установке модуля DQF

Москва 2025



Оглавление

1	Журнал изменений	3
2	Введение	4
2.1	Термины и определения	4
2.2	Сокращения и обозначения	5
2.3	Общие положения	5
3	Назначение ПО	7
4	Требования к установке	8
5	Подготовка к установке	9
6	Установка ПО	10
6.1	Установка без доступа к репозиторию	12
6.2	Порты доступа к сервисам ADC DQF:	13
7	Проверка доступности сервиса	14
8	Обновление ПО	15
9	Логирование сервисов модуля ADC DQF	16
10	Мониторинг модуля ADC DQF	17
10.1	Метрики	17
11	Keycloak	18
11.1	Интеграция ADC DQF и Keycloak	18
11.2	HTTPS для Keycloak	20
12	Компоненты дистрибутива	22
13	Контакты технических специалистов	23

2 Введение

2.1 Термины и определения

Термин	Значение
Application Programming Interface (API)	Программный интерфейс приложения, интерфейс прикладного программирования – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
Data Quality Framework (DQF)	Программное обеспечение, позволяющее производить проверки данных с использованием настраиваемых правил, сконфигурированных в соответствии с особенностями предметной области и бизнес-логики заказчика.
Kafka	Apache Kafka — распределённый программный брокер сообщений с открытым исходным кодом
Keycloak	Инструмент с открытым исходным кодом, используемый для аутентификации пользователей в организации, с возможностью настройки single sign-on и управления доступом
SQL	Декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей СУБД
База данных	Совокупность данных, хранимых в соответствии со схемой, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных
Задача	Процесс создания проверок по группе правил для выявления несоответствий или подтверждения качества исходных данных .
Правило (Rule)	Структурированная последовательность шагов (базовых алгоритмов, механизмов управления потоком и других правил), где каждый шаг возвращает статус выполнения и оперирует данными через контекст исполнения для решения конкретной бизнес-задачи.
Проверка	Процесс исполнения заданного правила над исходными данными для выявления несоответствий или подтверждения их качества.
Сервис	Единица развертывания, реализующая конкретный набор услуг/функций в рамках ПО DQF
Сервис каталога правил (СКП)	Сервис, позволяющий пользователю управлять правилами и группами правил.
Сервис исполнения правил (СИП)	Сервис для реализации выполнения задач по автоматической верификации данных с использованием заданного правила.

Сервис планирования задач (СПЗ)	Сервис создания задач для проверки набора данных и формирующий сообщения-запросы для Сервиса исполнения правил.
Сервис расписания задач (СРЗ)	Сервис для планирования и создания задач для проверки данных по определенному расписанию.

2.2 Сокращения и обозначения

Сокращение	Наименование
ПО	Программное обеспечение
СУБД	Система управления базами данных
БД	База данных
AD.C	Arenadata Catalog
ADC DQF	Модуль Data Quality Framework (DQF) Arenadata Catalog
УЗ	Учетная запись
ТП	Техническая поддержка

2.3 Общие положения

Инструкция предназначена для должностных лиц, осуществляющих установку программного обеспечения модуля DQF Arenadata Catalog (ADC). Документ содержит пошаговое описание действий для установки специального программного обеспечения системы.

Модуль Arenadata Catalog Data Quality Framework (далее- ADC DQF) обеспечивает возможность развертывания с использованием контейнерной архитектуры. Предоставляемый клиентам пакет дистрибутива модуля ADC DQF включает набор компонентов необходимый для полноценного развертывания решения, таблица 1.

Таблица 1 Компоненты модуля ADC DQF

Функциональный блок	Компоненты продукта
Сервис исполнения правил (СИП)	DQF Engine Standalone
Сервис расписаний задач (СРЗ)	DQF Scheduler
Сервис планирования задач (СПЗ)	DQF Planner
Сервис каталога правил (СКП)	DQF CAS
Сервис результатов	DQF Aggregator
Сервис дашбордов	DQF Dashboards Storage
Сервис отчетов	DQF Reporter
Веб интерфейс	DQF Web App

Компоненты, дополнительно необходимые для эксплуатации модуля ADC DQF, описаны в таблице 2.

Таблица 2 Дополнительные компоненты для среды модуля ADC DQF

Функциональный блок	Компонент среды функционирования
Служебная база данных	Postgres 14 или любая совместимая с ней СУБД
Брокер сообщений	Kafka (предпочтительнее), RabbitMQ
Средство аутентификации	KeyCloak - 23.0
Перенаправление логов	Vector - 0.28.0
Хранение и поиск логов	Loki - 2.6.0, Grafana - 9.5.5
Сбор метрик	Prometheus - 2.41.0, Prometheus Graphite Exporter - 0.13.1
Балансировщик нагрузки	Nginx

3 Назначение ПО

Модуль Data Quality Framework ADC (далее – ADC DQF)- высокопроизводительный сервис выполнения проверок качества данных по настраиваемым правилам с консолидацией результатов и формированием инцидентов качества.

Функциональные возможности ADC DQF позволяют с помощью специальных правил и алгоритмов производить верификацию проверяемого объекта и его атрибутов и выявлять ошибки, а также настраивать проверки данных в соответствии с предметной областью и заданной конфигурацией (бизнес-логики) исполняемой проверки без доработки программного обеспечения.

ADC DQF позволяет обеспечить:

- соблюдение критериев качества данных;
- исполнение требований регуляторов;
- поддержку процессов цифровизации с применением методик Data Quality на новых платформах;
- соблюдение критериев качества данных при миграции и консолидации данных, предобработке данных;
- проактивное реагирование на инциденты качества и аномалии данных;
- централизацию проверок, реализацию методологий Data Quality и Data Governance в рамках одного интерфейса.

4 Требования к установке

Для установки ADC DQF по данной инструкции необходимо выполнение следующих требований:

1. Операционная система– Ubuntu 22.04.4 LTS, CentOS 7.9.2009, Ред ОС 7.3, ALT Server 10.1 (Mendeleevium), Astra Linux Воронеж 1.7.
2. ПО Docker version 25.0.0, и выше
3. ПО Docker compose version 1.29.2, и выше
4. Пользователь с правами sudo.(или пользователь добавленный в группу docker)

Следует проверить версию установленного Docker Compose на соответствие требованиям. В неподходящей версии Docker compose сборка не произойдет, так как используется версия контейнера 3.9. В случае отсутствия необходимо произвести установку согласно инструкции по [ССЫЛКЕ](#).

5 Подготовка к установке

Необходимо выполнить команду аутентификации на сервере репозитория, используя реквизиты, полученные в технической поддержке:

```
docker login repo.arenadc.io -u login -p password
```

Далее необходимо загрузить файл Docker compose и environment файлы по ссылке [предоставленной технической поддержкой](#).

В состав поставки ADC DQF не входит Keycloak. По умолчанию используется файл docker-compose.yaml

В случае необходимости подключения модуля ADC DQF к установленному Keycloak следует использовать файл Docker compose docker-compose-keycloak.yaml. В этом случае необходимо выполнить настройку с поставляемыми файлами окружения.

В environment файлах:

- dqf.json – отвечает за наполнение realm хранилища Keycloak модуля аутентификации, необходимо изменить поля связанные с host, на котором развернуто приложение,
- ragam-keycloak.env – отвечает за параметризацию keycloak, необходимо изменить IP адреса в переменных окружения, отвечающих за аутентификацию через keycloak:

```
DQF_SPRING_PROFILES=keycloak, dqf-kafka
PUBLIC_PROFILES_ACTIVE=keycloak
PUBLIC_KEYCLOAK_URL=https://localhost/keycloak
ES_REPLICAS=2
KC_HOSTNAME_URL=https://localhost/keycloak
KC_HOSTNAME_ADMIN_URL=https://localhost/keycloak
```

Указанные выше файлы необходимы для параметризации Docker compose и интеграции Keycloak с ADC DQF. Есть альтернативный способ настройки интеграции через web-интерфейс Keycloak, который находится на порту 8081.

Для загрузки набора Правил для демонстрации функциональных возможностей ADC DQF в Каталог используется отдельный файл Docker compose upload-demo-rules.yaml

6 Установка ПО

Для работы приложения рекомендуется использовать директорию “/opt/adc_dqf”.

Загруженную конфигурацию для docker compose поместить в директорию “/opt/adc_dqf/” и перейти в эту директорию.

```
cd /opt/adc_dqf/
```

Для обеспечения функции логирования для сервисов модуля ADC DQF требуется выдать права на запись логов

```
sudo -v
sudo chmod -R 0777 aggregator/logs
sudo chmod -R 0777 cas/logs
sudo chmod -R 0777 dashboards-storage/logs
sudo chmod -R 0777 engine/logs
sudo chmod -R 0777 engine/probe-logs
sudo chmod -R 0777 planner/logs
sudo chmod -R 0777 reporter/logs
sudo chmod -R 0777 scheduler/logs
sudo chmod -R 0777 aggregator/data
sudo chmod -R 0777 cas/data
sudo chmod -R 0777 dashboards-storage/data
sudo chmod -R 0777 planner/data
sudo chmod -R 0777 reporter/data
sudo chmod -R 0777 scheduler/data
```

!Лицензия на модуль ADC DQF

Лицензию `dqf.license` следует скопировать /opt/adc_dqf/engine/

По вопросу получения лицензии обращайтесь в технической поддержку, используя электронный адрес info@arenadc.io

Настройка конкурентности Сервиса исполнения правил (engine). СИП позволяет настраивать конкурентность исполнения правил горизонтально и вертикально:

- Настройка горизонтальной конкурентности

Предназначен в первую очередь для повышения отказоустойчивости модуля ADC DQF. Выполняется путем настройки количества инстансов (реплик) сервиса в docker compose файле:

```
  dqf-es:
    image: repo.arenadc.io/dqf-internal-release/dqf/engine-standalone:${ES_VERSION}
    <<: *boot-common
    ports:
      - '8315-8320:8080'
      - '1020-1030:9020'
    environment:
      <<: *boot-env
      LOGGING_THRESHOLD_FILE: INFO
      LOGGING_FILE_NAME: /logs/dqf-engine-standalone.log
      JAVA_TOOL_OPTIONS: |
```

```
-Xmx500m
volumes:
  - './engine/logs:/logs:rw'
  - './engine/test-dqf-es.license:/app/dqf-engine.licence:ro'
deploy:
  replicas: 2
  endpoint_mode: vip
  resources:
    limits:
      cpus: "1.0"
      memory: 500m
  healthcheck:
    <<: *boot-health
  test: wget --proxy off -T5 -qO- http://localhost:8080/engine-
api/actuator/health | grep UP || exit 1
```

- Настройка вертикальной конкурентности

Предназначен для оптимизации быстродействия выполнения правил СИП путем изменения количества одновременно обрабатываемых потоков. Параметр связан с конфигурацией Kafka – количество партиций. Значение количества потоков устанавливается в конфигурационном файле СИП application-dqf-kafka.yaml:

```
spring:
  kafka:
    bootstrap-servers: kafka-0:9092
    concurrency: 1
    consumer:
      group-id: dqf-es
      auto-offset-reset: earliest
  dqf:
    controller:
type: KAFKA
```

Настройка кэширования Правил для СИПа

Предназначена для оптимизации взаимодействия с сервисом Каталог правил и быстродействия выполнения правил СИП. Значение количества последних используемых Правил сохраненных в кэше устанавливается в конфигурационном файле СИП application.yaml:

```
c engine:
  algorithms-cache:
    enabled: true
    maximum-size: 100
    expire-after-write: PT30S
  data-store:
    mem:
enabled: true
```

Чтобы запустить ПО выполните в консоли одну из следующих команд:

```
sudo docker compose -f docker-compose.yaml up -d
```

или в случае подключения модуля ADC DQF к установленному Keycloak

```
sudo docker compose -f docker-compose-keycloak.yaml --env-file ./env/demo_stand.env up -d
```

Установка сервисов модуля ADC DQF выполняется в пределах нескольких минут.

Если после установки UI модуля ADC DQF недоступен, то возможно не успел запуститься nginx. В этом случае необходимо выполнить повторный запуск nginx в ручном режиме. Для этого необходимо выполнить следующую команду:

```
sudo docker compose -p docker-compose-simplified up -d nginx
```

Конфигурирование сервисов модуля ADC DQF возможно несколькими способами: docker compose, .env и config server. По умолчанию в поставляемом ADC DQF используется config sever. Сами конфигурации находятся в каталоге engine/config/ соответствующего сервиса ADC DQF.

В случае подключения модуля ADC DQF к установленному Keycloak, конфигурация подключения к Keycloak указана в compose файле.

Сервисы ADC DQF используют внутреннюю БД PostgreSQL (версия 14.x). Разворачивание БД сконфигурировано в Docker compose файле.

В случае необходимости изменить конфигурации сервисов ADC DQF необходимо остановить контейнер и снова запустить Docker compose файл с измененной конфигурацией (вариант без Keycloak)

```
docker compose -p docker-compose-simplified down  
docker compose -p docker-compose-simplified up -d
```

6.1 Установка без доступа к репозиторию

Скачать образы приложения по предоставленным ссылкам.

Или выгрузить образы в архив локально:

```
docker save -o your_archive_name.zip repo_image_name:image_tag
```

Загрузить скаченные образы в docker images:

```
docker load < adc-dqf-vx.x.x.zip
```

После загрузки всех образов выполнить установку приложения согласно шагам, описанным в п. 6.

6.2 Порты доступа к сервисам ADC DQF:

По умолчанию сервисы модуля ADC DQF доступны по следующим портам:

- DQF Web App (Веб интерфейс)
* url: <http://localhost:1000>
- DQF Planner (API Планировщика)
* swagger: <http://localhost:8080/planner-api/swagger-ui/index.html>
* actuator: <http://localhost:8080/planner-api/actuator>
- DQF CAS (API Каталога правил)
* swagger: <http://localhost:8081/cas-api/swagger-ui/index.html>
* actuator: <http://localhost:8081/cas-api/actuator>
- DQF Scheduler (API Сервиса расписаний)
* swagger: <http://localhost:8082/scheduler-api/swagger-ui/index.html>
* actuator: <http://localhost:8082/scheduler-api/actuator>
- DQF Aggregator (API Агрегатора)
* swagger: <http://localhost:8083/aggregator-api/swagger-ui/index.html>
* actuator: <http://localhost:8083/aggregator-api/actuator>
- DQF Reporter (API Сервиса отчетов)
* swagger: <http://localhost:8084/reporter-api/swagger-ui/index.html>
* actuator: <http://localhost:8084/reporter-api/actuator>
- DQF Dashboards Storage (API Дашборды)
* swagger: <http://localhost:8085/dashboards-storage-api/swagger-ui/index.html>
* actuator: <http://localhost:8085/dashboards-storage-api/actuator>
- Kafka
* ui: <http://localhost:9080>
* exporter: <http://localhost:9308>
- Prometheus
* url: <http://localhost:3100>
- Grafana
* url: <http://localhost:3000>
* username: `admin`
* password: `admin`

Порты по умолчанию настраиваются в Docker compose файле.

Доступы к БД сервисов ADC DQF также настраиваются в Docker compose файле.

7 Проверка доступности сервиса

Понять о состоянии сервисов можно посредством команд:

```
docker ps -a
docker compose ps -a
docker compose logs <container name>
```

Также состояние каждого из сервисов модуля ADC DQF можно проверить используя API актуатора соответствующего сервиса.

После инициализации сервисов и БД запускаем модуль ADC DQF (без подключения keycloak) в web-браузере по адресу:

```
http://[server_address]:1000
```

В случае технической необходимости остановить сервисы и/или переустановить сервисы используем следующие команды:

```
cd /opt/adc #переход в рабочую директорию
docker compose "chosen env files" stop #остановить все контейнеры
docker volume prune #очистить все тома приложения
docker container prune #удалить остановленные контейнеры
```

8 Обновление ПО

Обновление приложения может быть произведено следующим образом.

- Отключить старую версию программы выполнив:

```
sudo docker compose -p docker-compose-simplified down
```

- Загрузить новую версию ПО (файлы окружения, образы) по предоставленным ссылкам.
- Изменить конфигурационные файл.
- Включить приложение (п. 6 инструкции).

9 Логирование сервисов модуля ADC DQF

Сервисы модуля ADC DQF выполняет логирование с записью логов с настроенным уровнем детализации в соответствующий файл.

Уровень логирования, может быть, один из следующих:

- TRACE- самый детальный уровень логирования;
- DEBUG – логирование всех видов событий;
- INFO – логирование ошибок, предупреждений и сообщений;
- WARN – логирование ошибок и предупреждений;
- ERROR – логирование всех ошибок.

Настройка уровня логирования и расположения файла с логами выполняется в Docker compose файле. Настройка имеет следующий вид:

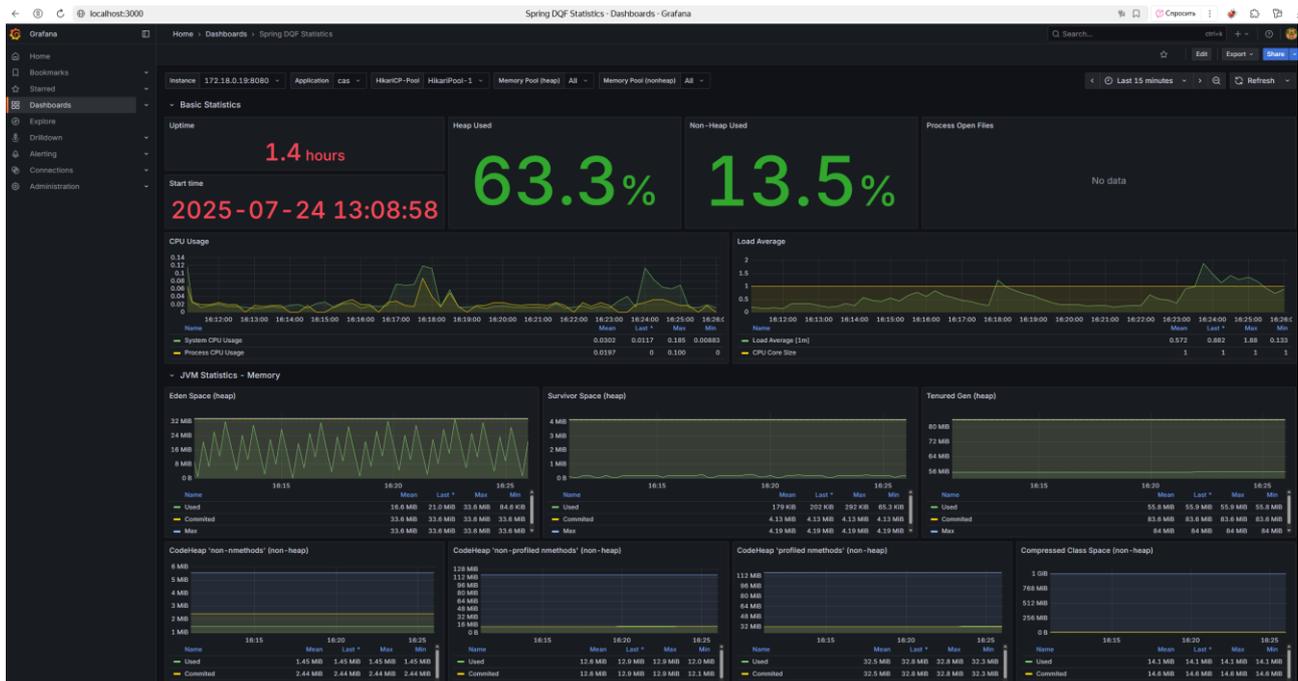
```
dashboards-storage:
  image: repo.arenadc.io/dqf-internal-release/dqf/dashboards-
storage:latest
  container_name: dqf-dashboards-storage
  <<: [ *boot-common, *docker-logging ]
  environment:
    <<: *boot-env
    # Настройки логирования для записи в файл
    LOGGING_THRESHOLD_FILE: TRACE
    LOG_FILE: /app/log/dqf-aggregator.log
```

10 Мониторинг модуля ADC DQF

10.1 Метрики

ADC DQF поставляется с настроенными дашбордами в графене. Доступны стандартные метрики Prometheus Spring- дашборд Spring DQF Statistics.

Так же в Графене есть одна витрина с метриками по SQL запросам:



Статистика отображается отдельно по каждому инстансу (сервису)
Доступны метрики:

1. JVM - Micrometer – метрики JVM – heap memory, GC, операции ввода/вывода и тд.
2. HikariCP Statistics
3. HTTP Statistics
4. Tomcat Statistics
5. Logback Statistics

11 Keycloak

11.1 Интеграция ADC DQF и Keycloak

Создать realm:

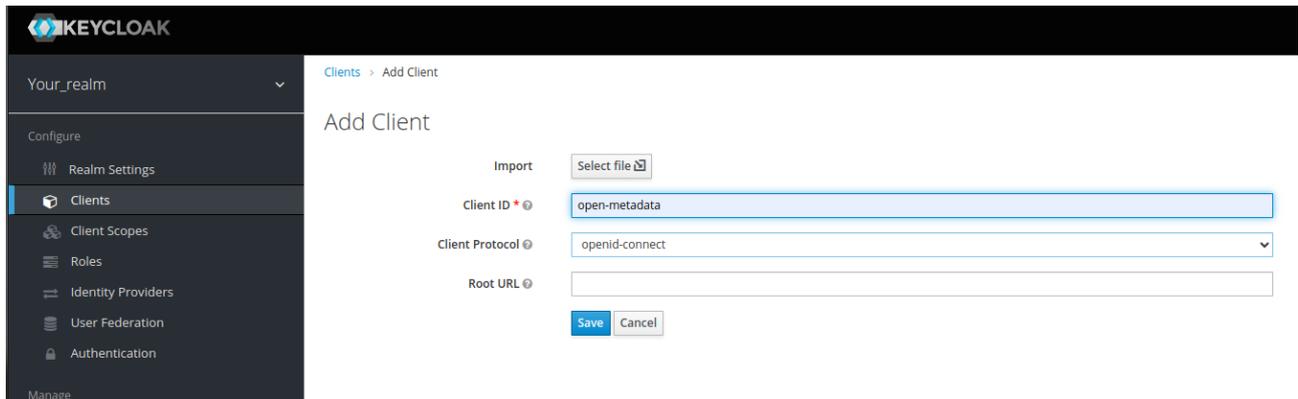
Add realm

Import

Name *

Enabled

Добавить Client:



KEYCLOAK

Your_realm

Configure

- Realm Settings
- Clients**
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

Clients > Add Client

Add Client

Import

Client ID *

Client Protocol

Root URL

Задать и сохранить настройки клиента:

Settings | Keys | Roles | Client Scopes | Mappers | Scope | Revocation | Sessions | Offline Access | Installation

Client ID

Name

Description

Enabled

Always Display in Console

Consent Required

Login Theme

Client Protocol

Access Type

Standard Flow Enabled

Implicit Flow Enabled

Direct Access Grants Enabled

Service Accounts Enabled

OAuth 2.0 Device Authorization Grant Enabled

OIDC CIBA Grant Enabled

Authorization Enabled

Front Channel Logout

OIDC CIBA Grant Enabled OFF

Authorization Enabled OFF

Front Channel Logout OFF

Root URL

* Valid Redirect URIs +

Base URL

Admin URL

Logo URL

Policy URL

Terms of service URL

Web Origins +

Backchannel Logout URL

Backchannel Logout Session Required ON

Backchannel Logout Revoke Offline Sessions OFF

Создать user:

Add user

ID

Created At

Username *

Email

First Name

Last Name

User Enabled ON

Email Verified OFF

Groups No group selected

Required User Actions

Задать пользователю пароль, снять флажок Temporary:

Users > admin

Admin

Details Attributes **Credentials** Role Mappings Groups Consents Sessions

Manage Credentials

Position	Type	User Label	Data	Actions
----------	------	------------	------	---------

Set Password

Password

Password Confirmation

Temporary OFF

Поменять переменные окружения `demo_stand.env` (смотри раздел 5 данного документа), указав хост и порт к своему keycloak и новому realm (`Dqf.json`).

Перезапустить контейнеры ADC DQF ((смотри раздел 6 данного документа).

11.2 HTTPS для Keycloak

При использовании IP адресов, соответствующих маске внешних IP, Keycloak для работы требует подключение по зашифрованному протоколу `https`. Для этого в контейнер Keycloak нужно передать TLS сертификаты. В данном примере используется самоподписанный сертификат, также можно использовать доверенные сертификаты.

1. Обновить `docker-compose` файл - раздел с контейнером кейклок следующим образом:

```
keycloak:
  image: ${IMG_REPO}/keycloak:${KEYCLOAK_ARTIFACT_VERSION}
  command:
    - start-dev
    - --import-realm
  environment:
    KEYCLOAK_IMPORT: /tmp/realm-export.json -
Dkeycloak.profile.feature.upload_scripts=enabled
    KEYCLOAK_ADMIN: ${KEYCLOAK_CONSOLE_ADMIN}
    KEYCLOAK_ADMIN_PASSWORD: ${KEYCLOAK_CONSOLE_ADMIN_PASSWORD}
    KC_HTTPS_CERTIFICATE_FILE: /etc/x509/https/certificate.crt
    KC_HTTPS_CERTIFICATE_KEY_FILE: /etc/x509/https/private.key
    KC_PROXY: edge - при использовании прокси
  ports:
    - "8081:8080"
    - "8843:8443"
  volumes:
    - ./config/data-sec.json:/opt/keycloak/data/import/data-sec.json
    - ./keycloak/keycloak.conf:/opt/keycloak/conf/keycloak.conf
    - keycloak-data:/opt/keycloak/data:rw
    - ./certificate.crt:/etc/x509/https/certificate.crt
    - ./private.key:/etc/x509/https/private.key
  networks:
    local_app_net:
```

2. Создать конфигурационный файл для сертификата (файл `req.conf`):

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
C = US
ST = VA
L = SomeCity
O = MyCompany
OU = MyDivision
CN = YOUR_IP
[v3_req]
keyUsage = keyEncipherment, dataEncipherment, digitalSignature
```

```
extendedKeyUsage = serverAuth
subjectAltName = @alt_names
[alt_names]
IP.1 = YOUR_IP
```

3. Создать ключ и сертификат следующей командой:

```
openssl req -newkey rsa:2048 -nodes -keyout private.key -x509 -days 365 -
out certificate.crt -config req.conf -extensions 'v3_req'
```

4. Если сертификат не является доверенным добавить в JAVA Trusted Store приложения:

- Скопировать рутовый TLS сертификат в контейнер `adc_server`:

```
docker cp certificate_file adc_server_1:/adc/conf
```

- Зайти в контейнер `adc_server`:

```
docker exec -it adc_server bash
```

- Перейти в директорию с Java Trusted Store

```
cd /usr/lib/jvm/jdk-17.0.9-bellsoft-x86_64/lib/security/
```

- Там будет файл: `cacerts` это и будет Java Trusted Store.
- Загрузить сертификат в Java Trusted Store

```
keytool -import -alias CHOOSE-YOUR-ALIAS -file
/adc/conf/YOUR_CERTIFICATE_FILE -keystore /usr/lib/jvm/jdk-17.0.9-
bellsoft-x86_64/lib/security/
Пароль от trusted store: changeit
```

- Проверить наличие сертификата в Trusted Store по заданому ранее алиасу

```
keytool -list -keystore /usr/lib/jvm/jdk-17.0.9-bellsoft-
x86_64/lib/security/ , так же потребуется ввод пароля: changeit
```

- Перезапустить модуль ADC DQF через `docker stop/start`.

12 Компоненты дистрибутива

Функциональный блок	Компонент платформы
Сервис исполнения правил (СИП)	Engine
Служебная база данных	PostgreSQL - 14.6
Сервис каталога правил (СКП)	CAS
Загрузчик правил с набором демо правил	CAS updater
Сервис планирования задач (СПЗ)	Planner
Сервис расписаний задач (СРЗ)	Scheduler
Сервис результатов (СР)	Aggregator
Сервис отчетов (СО)	Reporter
Сервис дашбордов (СД)	Dashboards-storage
Сервис сообщений	Apache Kafka
Вэб сервер- балансировщик нагрузки	nginx
Postman коллекция	postman
Хранение метрик	Grafana - 9.5.5
Сбор метрик	Prometheus - 2.41.0
Средство аутентификации (опционально)*	KeyCloak - 23.0

demo_stand.env — файл с переменными окружения для аутентификации в приложении с keycloak

Dqf.json — конфигурация realm в keycloak

docker-compose-simplified-keycloak.yaml — docker compose для запуска приложения с keycloak

docker-compose-simplified.yaml — docker compose для запуска приложения без keycloak

grafana — настройки grafana, дашборд

prometheus — конфигурационный файл prometheus

13 Контакты технических специалистов

В случае возникновения трудностей при установке программного обеспечения, свяжитесь с технической поддержкой, используя электронный адрес info@arenadc.io.