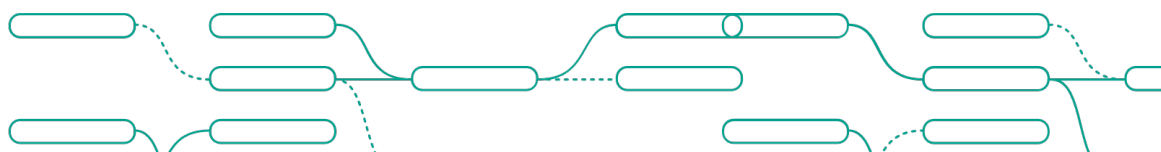


# Arenadata Catalog

Инструкция по установке модуля ADC.DQF

Москва 2026



## Оглавление

<b>1</b>	<b>Введение</b> .....	<b>3</b>
1.1	Термины и определения .....	3
1.2	Сокращения и обозначения .....	4
1.3	Общие положения.....	4
<b>2</b>	<b>Назначение ПО</b> .....	<b>6</b>
<b>3</b>	<b>Требования к установке</b> .....	<b>7</b>
<b>4</b>	<b>Подготовка к установке</b> .....	<b>8</b>
<b>5</b>	<b>Установка ПО</b> .....	<b>9</b>
<b>6</b>	<b>Проверка доступности сервиса</b> .....	<b>11</b>
<b>7</b>	<b>Обновление ПО</b> .....	<b>12</b>
<b>8</b>	<b>Логирование сервисов модуля ADC.DQF</b> .....	<b>13</b>
<b>9</b>	<b>Мониторинг модуля ADC.DQF</b> .....	<b>14</b>
10.1	Метрики .....	14
<b>10</b>	<b>Настройки авторизации</b> .....	<b>Ошибка! Закладка не определена.</b>
11.1	Интеграция ADC.DQF и Keycloak .....	<b>Ошибка! Закладка не определена.</b>
<b>11</b>	<b>Компоненты дистрибутива</b> .....	<b>15</b>
<b>12</b>	<b>Контакты технических специалистов</b> .....	<b>16</b>

# 1 Введение

## 1.1 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	Независимый модуль, реализующий конкретный набор услуг/функций в рамках модуля ADC.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 определённому расписанию.

## 1.2 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

## 1.3 Общие положения

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

Модуль ADC.DQF обеспечивает возможность развертывания с использованием контейнерной архитектуры. Предоставляемый клиентам пакет дистрибутива модуля ADC.DQF включает набор компонентов необходимый для полноценного развертывания решения. Состав пакета приведен в таблице 1.

**Таблица 1 Компоненты модуля ADC.DQF**

Функциональный блок	Компоненты продукта
Сервис исполнения правил (СИП)	DQF Engine Standalone
Сервис проверки правил (СПП)	DQF Engine Standalone (профиль probe)
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
Сервис авторизации	DQF Auth
Адаптер результатов для модуля сохранения	DQF Saver Adapter

Модули экосистемы Arenadata Catalog, которые дополняют ADC.DQF и обеспечивают дополнительную функциональность, описаны в таблице 2.

**Таблица 2 Компоненты экосистемы ADC**

Функциональный блок	Компоненты продукта
Модуль сохранения подробных результатов*	Result Saver
*обеспечивает функцию сохранения результатов в пользовательское s3 совместимое хранилище.	

Инфраструктурные компоненты, необходимые для эксплуатации ADC.DQF, описаны в таблице 3.

**Таблица 3 Дополнительные компоненты для среды модуля ADC.DQF**

Функциональный блок	Компонент среды функционирования
Перенаправление логов	Vector - 0.28.0
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
Единая точка входа	Nginx
Служба координации**	Zookeeper - 3.9
** используется для распространения и проверки лицензий	

## 2 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.

## 3 Требования к установке

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

1. Операционная система– Ubuntu 22.04.4 LTS, 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)

Перед началом установки следует проверить версию установленного Docker Compose на соответствие требованиям.

### Важно!

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 <https://docs.docker.com/engine/install/ubuntu/>

## 4 Подготовка к установке

Локальная среда настраивается в формате docker compose (docker-compose.yml)

Для установки модуля ADC.DQF необходимо подключиться к репозиториям [repo.arenadc.io](https://repo.arenadc.io), либо запросить архив с образами у службы поддержки Arenadata Catalog, направив соответствующий запрос на адрес [info@arenadc.io](mailto:info@arenadc.io).

Пример команды shell:

```
docker login repo.arenadc.io
```

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

Общий вид подготовленного каталога стенда должен иметь вид:

```
.
├── README.md
├── client
│   ├── postman
│   └── scripts
├── docker
│   ├── abstract
│   ├── misc
│   ├── modules
│   ├── profile-dev
│   └── profile-stand
└── scripts
    ├── README.md
    ├── backup.sh
    ├── migrate-mounts.sh
    └── release
```

Запросить у службы поддержки лицензию [dqf.license](#). Указанный файл требуется положить (`.docker/modules/zk/license/es.adc.license`).

Полученный архив с образами должен быть расположен в каталоге: `docker/profile-stand/image/`

## 5 Установка ПО

Для установки ПО модуля ADC.DQF требуется выполнить скрипт инициализации для чего перейти в каталог `cd ./docker/profile-stand` и запустить скрипт инициализации `./init_stand.sh`

В процессе выполнения скрипт запросит название стенда (например: `demo`) и его IP-адрес (например: `10.80.0.70`). Если существует каталог с именем `stand-<stand_name>`, где `<stand_name>` - указанное имя, скрипт запросит у пользователя разрешение на перезапись конфигурационных файлов стенда (если пользователь запретит, скрипт пропустит этот шаг).

Скрипт инициализации выполняет следующие шаги:

- Генерирует ключ и сертификат стенда по пути `./certs/stand`.
- Скачивает необходимые драйвера для работы с БД.
- Подготавливает необходимые директории для маунтов (БД, логи).
- Создает файлы:
  - Настроенные переменные `docker compose`: `stand-<stand_name>/ .env`.  
Например: `stand-demo/.env`
  - `docker-compose`-файл со специфичными для стенда сервисами: `stand-<stand_name>/docker-compose-stand-specific.yaml`  
он позволяет как добавлять новые сервисы, так и перекрывать уже существующие (добавлением профиля, можно их исключать из запуска).  
Например: `stand-demo/docker-compose-stand-specific.yaml`
  - Пустой файл конфигурации, куда можно добавить специфичные для стенда настройки:  
`stand-<stand_name>/spring-common/application.yaml`  
Например: `stand-demo/spring-common/config/application.yaml`
  - Markdown-файл с валидными ссылками на сервисы: `stand-<stand_name>/services.md`  
Например: `stand-demo/services-demo.md`.
  - Локальный `env`-файл для запуска `docker compose` для конкретного стенда: `.env`  
Используется в точке запуска: `redeploy-stand.sh`

Для всех сервисов модуля ADC.DQF в соответствии с `docker compose` определены специфичные конфигурационные файлы в соответствующих каталогах сервисов и общие для всех сервисов в каталоге `stand-<stand_name>/spring-common` (`stand-demo/spring-common`):

- ``application.yaml``;
- ``application-dqf-kafka.yaml``;
- ``application-keycloak.yaml``.

В состав поставки ADC.DQF входит свой инстанс Keycloak. В случае необходимости подключения модуля ADC.DQF к уже используемому Keycloak или Avanpost требуется в сформированный файл `docker/profile-stand/stand-demo/spring-common/config/application.yaml` добавить настройки подключения к соответствующему Keycloak.

### Внимание!

Файл **`application-keycloak.yaml`** изменять не нужно!

(`docker/profile-stand/stand-demo/spring-common/config/application-keycloak.yaml`)

При подключении к уже используемому Keycloak требуется создать/загрузить в realm данные по пользователям и атрибутам доступа для аутентификации и авторизации в модуле ADC.DQF используя файл `Dqf.json` (`docker/modules/keycloak/Dqf.json`)

После отработки скрипта инициализации формируется новая точка запуска `redeploy-stand.sh`.

```
#!/bin/bash

docker compose -f docker-compose-main.yaml --env-file .env stop
docker compose -f docker-compose-main.yaml --env-file .env up -d --pull always
```

При старте среды в Сервис каталога правил (DQF CAS) загружаются демо-правила (`docker/modules/cas-updater/algorithms/demo`).

Установка сервисов модуля ADC.DQF выполняется в пределах нескольких минут.

Web-интерфейс (DQF Web App) запускает фронт-энд приложение графического конструктора проверок и Nginx в качестве reverse прокси, чтобы front-end приложение и back-end сервисы работали и были доступны под одним доменом.

Если после установки UI модуля ADC.DQF недоступен, то, возможно, не успел запуститься nginx или Keycloak. В этом случае необходимо дождаться запуска контейнера Keycloak, затем выполнить повторный запуск (или перезапуск) контейнера nginx в ручном режиме. Для этого необходимо выполнить следующую команду:

```
docker compose -f docker-compose-main.yaml --env-file .env up -d nginx
```

для перезапуска

```
docker restart nginx-reverse-proxy
```

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

Если нужно просто обновить контейнеры на новые образы, достаточно запустить `redeploy-stand.sh`. Так как в `docker-compose` указаны `latest`-образы и `up` вызывается с параметром `--pull=always`, будут скачаны самые последние версии образов.

В случае необходимости изменить конфигурации сервисов ADC.DQF необходимо остановить контейнер и снова запустить Docker compose файл с измененной конфигурацией

```
docker compose -f docker-compose-main.yaml --env-file .env down
docker compose -f docker-compose-main.yaml --env-file .env up -d
```

## 6 Проверка доступности сервиса

Информацию о состоянии сервисов можно получить, выполнив команды:

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

После инициализации сервисов и БД запускаем модуль ADC.DQF в web-браузере по адресу:

```
https://[server_address]
```

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

```
docker compose -f docker-compose-main.yaml --env-file .env down
docker compose -f docker-compose-main.yaml --env-file .env up -d
```

## 7 Обновление ПО

Обновление приложения может быть произведено следующим образом:

- Отключить старую версию программы выполнив:

```
docker compose -f docker-compose-main.yaml --env-file .env down
```

- Загрузить новую версию ПО (файлы окружения, образы) по предоставленным ссылкам.
- Выполнить подготовку к установке новой версии, включая настройки конфигурации, согласно раздела 5 настоящего документа.
- Выполнить установку и запустить модуль ADC.DQF, согласно раздела 6 настоящего документа.

## 8 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 для всех сервисов модуля ADC.DQF выполняется в файле `application.yaml` в каталоге `[spring-common]`:

```
logging:
  level:
    io:
      arenadc:
        dqf:
          service: info
```

Логи сохраняются в директории вида :

- `docker/profile-stand/mnt/logs/<service>`

например:

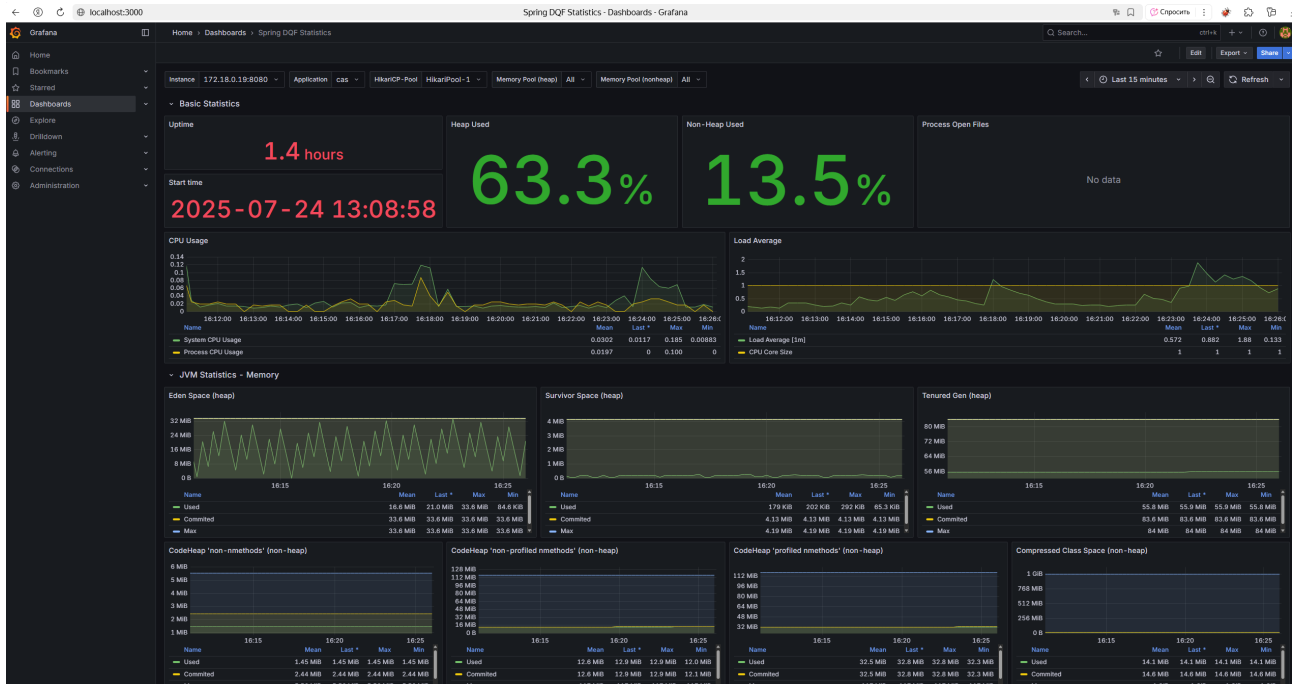
- `docker/profile-stand/mnt/logs/aggregator`
- `docker/profile-stand/mnt/logs/auth`
- `docker/profile-stand/mnt/logs/cas`
- `docker/profile-stand/mnt/logs/dashboards-storage`
- `docker/profile-stand/mnt/logs/engine`
- `docker/profile-stand/mnt/logs/engine-probe`
- `docker/profile-stand/mnt/logs/keycloak`
- `docker/profile-stand/mnt/logs/planner`
- `docker/profile-stand/mnt/logs/reporter`
- `docker/profile-stand/mnt/logs/result-saver`
- `docker/profile-stand/mnt/logs/result-saver-adapter`
- `docker/profile-stand/mnt/logs/scheduler`

## 9 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

## 10 Компоненты дистрибутива

Функциональный блок	Компонент платформы
Сервис исполнения правил (СИП)	DQF Engine Standalone
Сервис проверки правил (СИП)	DQF Engine Standalone (профиль probe)
Сервис расписаний задач (СРЗ)	DQF Scheduler
Сервис планирования задач (СПЗ)	DQF Planner
Сервис каталога правил (СКП)	DQF Complex Algorithms Storage Service
Сервис результатов	DQF Aggregator
Сервис дашбордов	DQF Dashboards Storage
Сервис отчетов	DQF Reporter
Веб-портал	DQF Web App
Сервис авторизации	DQF Auth Service
Адаптер результатов для модуля сохранения	DQF Saver Adapter
Модуль сохранения подробных результатов*	Result Saver
Службная база данных	Postgres 14 или совместимая
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
Прокси	Nginx
Служба координации	Zookeeper - 3.9

## 11 Контакты технических специалистов

В случае возникновения трудностей при установке программного обеспечения, свяжитесь с технической поддержкой, используя электронный адрес [info@arenadc.io](mailto:info@arenadc.io).