

# **Руководство по развертыванию и установке программного обеспечения**

Data Quality Framework

**Аннотация**

Настоящий документ описывает процесс развертывания и установки программного обеспечения (далее – ПО) Data Quality Framework (далее – ПО DQF).

Исключительные права на ПО DQF принадлежат ООО «Клин Дейта».

Оглавление

1. Общие сведения.....	5
2. Разворачивание программного обеспечения.....	6
2.1. Разворачивание компонента контроля качества (ПО DQF: Engine) .....	6
2.1.1. Состав поставки.....	6
2.1.2. Необходимые программы и компоненты.....	6
2.1.3. Последовательность установки.....	6
2.1.4. Установка и настройка общесистемного ПО.....	6
2.1.4.1. Требования к аппаратной платформе.....	6
2.1.4.2. Требования к сетевой инфраструктуре .....	7
2.1.4.3. Назначение прав для пользователя, под которым будет проводиться установка и настройка ПО DQF: Engine.....	7
2.1.4.4. Установка и настройка программного окружения.....	7
2.1.5. Установка и настройка ПО DQF: Engine.....	7
2.1.5.1. Установка и запуск дистрибутива ПО DQF: Engine .....	7
2.1.5.2. Проверка работоспособности дистрибутива .....	8
2.1.6. Перезапуск.....	9
2.2. Разворачивание компонента сопоставления данных (ПО DQF: Federation) .....	10
2.2.1. Состав поставки.....	10
2.2.2. Необходимые программы и компоненты.....	10
2.2.3. Последовательность установки.....	10
2.2.4. Установка и настройка общесистемного ПО.....	10
2.2.4.1. Требования к аппаратной платформе .....	10
2.2.4.1.1. Конфигурация сервиса Узла.....	10
2.2.4.1.2. Конфигурация БД Узла.....	11
2.2.4.1.3. Конфигурация сервиса Центра.....	11
2.2.4.1.4. Конфигурация БД Центра.....	11
2.2.4.1.5. Конфигурация БД кэша Центра .....	12
2.2.4.2. Требования к сетевой инфраструктуре .....	12
2.2.4.3. Назначение прав для пользователя, под которым будет проводиться установка и настройка ПО DQF: Federation.....	12
2.2.4.4. Установка и настройка программного окружения.....	12
2.2.5. Установка и настройка ПО DQF: Federation.....	13
2.2.5.1. Установка и запуск дистрибутивов ПО DQF: Federation .....	13
2.2.5.1.1. Установка и запуск сервиса Узла.....	13
2.2.5.1.2. Установка и запуск сервиса Центра .....	14
2.2.5.2. Проверка работоспособности ПО DQF: Federation.....	14
2.2.6. Перезапуск.....	16



## **1. Общие сведения**

Программное обеспечение Data Quality Framework (далее – ПО DQF) реализует функции контроля качества и сопоставления данных.

Компонент контроля качества (далее – ПО DQF: Engine) обеспечивает контроль данных, которые рассматриваются как целостное образование, состоящее из взаимосвязанных частей. ПО DQF: Engine нацелено на мониторинг качества данных на различных уровнях и обнаружение ошибок по сформированным правилам и критериям проверок.

Функциональные возможности ПО DQF: Engine 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.

Компонент сопоставления данных (далее – ПО DQF: Federation) обеспечивает сопоставление данных объектов (физических, юридических и иных объектов) из разных источников.

Компоненты ПО DQF могут работать как комплексно, так и независимо друг от друга.

## **2. Разворачивание программного обеспечения**

### **2.1. Разворачивание компонента контроля качества (ПО DQF: Engine)**

#### **2.1.1. Состав поставки**

В состав поставки ПО DQF: Engine входит дистрибутив ПО DQF: Engine для использования приобретаемого ПО как самостоятельно приложения.

Состав поставки, набор передаваемых прав, правила и порядок применения ПО DQF: Engine определяются лицензионным договором.

#### **2.1.2. Необходимые программы и компоненты**

Для установки и настройки дистрибутива ПО DQF: Engine требуется наличие следующего программного обеспечения:

- брокер сообщений (RabbitMQ);
- ПО Docker 20.10+;
- операционная система с возможностью установки ПО Docker 20.10+.

#### **2.1.3. Последовательность установки**

Для установки ПО DQF: Engine должны быть проведены следующие предварительные работы по подготовке инфраструктуры: произведена настройка сетевой инфраструктуры, программно-аппаратной платформы или настройка сервера приложений. Далее необходимо произвести установку ПО DQF: Engine.

Установка и администрирование ПО DQF: Engine осуществляются системным администратором, имеющим доступ к программно-аппаратной инфраструктуре, на которой будет произведена установка и осуществляться эксплуатация ПО DQF: Engine.

#### **2.1.4. Установка и настройка общесистемного ПО**

##### **2.1.4.1. Требования к аппаратной платформе**

Для установки ПО DQF: Engine необходимо обеспечить настройку сервера приложения в следующей конфигурации:

- Процессор:
  - Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);
  - от 4 ядер.
- ОЗУ:

- от 4 Гб.
- Дисковое пространство:
  - от 50 Гб.

#### **2.1.4.2. Требования к сетевой инфраструктуре**

Пропускная способность канала между сервером приложений и внешними информационными системами, направляющими запросы в ПО DQF: Engine, должна составлять не менее 1 Гбит/с.

Пропускная способность канала между сервером приложений и брокером сообщений должна составлять не менее 1 Гбит/с.

#### **2.1.4.3. Назначение прав для пользователя, под которым будет проводиться установка и настройка ПО DQF: Engine**

Для установки и настройки ПО DQF: Engine не требуется назначение прав пользователя, установку и настройку осуществляет системный администратор, имеющий доступ к серверу приложений, где будет установлено ПО DQF: Engine.

#### **2.1.4.4. Установка и настройка программного окружения**

После подготовки сетевой инфраструктуры и аппаратной платформы необходимо произвести работы по подготовке программного окружения для установки ПО DQF: Engine.

Для установки дистрибутива ПО DQF: Engine необходимо установить и предварительно настроить ПО Docker версии 20.10+ в соответствии с используемой операционной системой. Установку необходимо проводить, следуя инструкциям разработчика программного обеспечения. Инструкция для различных операционных систем доступна по ссылке: <https://docs.docker.com/engine/install/>.

В случае отсутствия в составе программного окружения брокера очередей, необходимо произвести установку, следуя инструкциям разработчика соответствующего программного обеспечения, доступных по ссылкам:

- ссылка на инструкцию по установке Rabbit: <https://www.rabbitmq.com/download.html>

#### **2.1.5. Установка и настройка ПО DQF: Engine**

##### **2.1.5.1. Установка и запуск дистрибутива ПО DQF: Engine**

Для установки дистрибутива ПО DQF: Engine предварительно необходимо скачать архив с предоставленным дистрибутивом в формате .tar и файл с шаблоном настроек,

необходимых для запуска ПО, и скопировать их на сервер, на котором должно быть развернуто ПО.

После чего в консоли управления сервером необходимо перейти в каталог со скаченным архивом и добавить дистрибутив ПО DQF: Engine в локальный Docker Registry, выполнив команду:

```
docker load -i dqf.tar
```

После чего, используя шаблон настроек, необходимых для запуска ПО, произвести настройку параметров запуска в зависимости от выбранного ПО брокера очередей. Ниже приведен пример настройки параметров запуска для брокера очередей RabbitMQ:

1. Необходимо создать файл `env.file` со следующими переменными окружения:

```
# spring.rabbitmq properties
SPRING_RABBITMQ_HOST: '' # fqdn или IP-адрес Rabbitmq
SPRING_RABBITMQ_PORT: 5672 # порт Rabbitmq
SPRING_RABBITMQ_USERNAME: '' # пользователь Rabbitmq
SPRING_RABBITMQ_PASSWORD: '' # пароль пользователя Rabbitmq
SPRING_RABBITMQ_VIRTUAL_HOST: '' # Rabbitmq Virtual Host, в котором
настроены используемые очереди
SPRING_RABBITMQ_TEMPLATE_EXCHANGE: '' # Rabbitmq exchange, связанный
(binding) с read и write очередями

# amqp.queue properties
AMQP_QUEUE_READ_NAME: '' # read очередь в Rabbitmq
AMQP_QUEUE_WRITE_NAME: '' # write очередь в Rabbitmq
```

2. Создать директорию `/etc/configurations` для хранения конфигураций комплексных алгоритмов на сервере приложений, где будет запущен дистрибутив DQF: Engine.

3. Произвести запуск ПО DQF: Engine в Docker используя команду:

```
docker run -d --env-file ./env.file --name dqf --mount
'source=/etc/configurations/,target=./configurations' dqf
```

#### 2.1.5.2. Проверка работоспособности дистрибутива

После завершения установки и настройки ПО необходимо выполнить проверку работоспособности ПО DQF: Engine. Для этого требуется направить тестовый запрос, используя установленный ранее брокер очередей RabbitMQ. Ниже приведен пример тестового запроса для брокера очередей RabbitMQ с описанием последовательности выполнения необходимых действий:

1. Зайти в консоль управления RabbitMQ.
2. Сформировать и отправить тестовый запрос в очередь, указанную в переменной `AMQP_QUEUE_WRITE_NAME` файла `env.file`, созданного в рамках п. 2.1.5.1 настоящего Руководства по развертыванию и установке.

3. Проверить, что направленный тестовый запрос обработан ПО DQF: Engine и результат исполнения проверки отправлен в очередь AMQP\_QUEUE\_READ\_NAME.

Описание форматов для формирования тестового запроса и проверки ответа приведено в документе «Руководство системного администратора программного обеспечения Data Quality Framework» в разделе 3.1.6.

### **2.1.6. Перезапуск**

Для перезапуска ПО DQF: Engine необходимо в Docker выполнить команду:

```
docker restart dqf
```

## **2.2. Разворачивание компонента сопоставления данных (ПО DQF: Federation)**

### **2.2.1. Состав поставки**

В состав поставки ПО DQF: Federation входят дистрибутивы следующих сервисов:

- узлы, обеспечивающие хранение свойств объекта, необходимых для сопоставления;
- центр, обеспечивающий управление поиском сопоставлений, формирование инцидентов качества данных, хранение результатов сопоставления.

Состав поставки, набор передаваемых прав, правила и порядок применения ПО DQF: Federation определяются лицензионным договором.

### **2.2.2. Необходимые программы и компоненты**

Для установки и настройки дистрибутивов ПО DQF: Federation требуется наличие следующего программного обеспечения:

- брокер сообщений Kafka;
- СУБД PostgreSQL 14+;
- СУБД Redis 7+;
- ПО Docker 20.10+;
- операционная система с возможностью установки ПО Docker 20.10+.

### **2.2.3. Последовательность установки**

Для установки ПО DQF: Federation должны быть проведены следующие предварительные работы по подготовке инфраструктуры: произведена настройка сетевой инфраструктуры, программно-аппаратной платформы или настройка сервера приложений.

Установка и администрирование ПО DQF: Federation осуществляются системным администратором, имеющим доступ к программно-аппаратной инфраструктуре, на которой будет произведена установка и осуществляться эксплуатация ПО DQF: Federation.

### **2.2.4. Установка и настройка общесистемного ПО**

#### **2.2.4.1. Требования к аппаратной платформе**

Для установки ПО DQF: Federation необходимо обеспечить настройку серверов приложений со следующей конфигурацией в зависимости от сервиса.

##### **2.2.4.1.1. Конфигурация сервиса Узла**

Процессор:

- Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);

- от 4 ядер.

ОЗУ:

- от 4 Гб.

Дисковое пространство:

- от 5 Гб.

#### **2.2.4.1.2. Конфигурация БД Узла**

Процессор:

- от 8 ядер.

ОЗУ:

- от 16 Гб.

Дисковое пространство:

- от 400 Гб.

#### **2.2.4.1.3. Конфигурация сервиса Центра**

Процессор:

- Intel Xeon Ice Lake или новее (Intel Xeon Silver 4310 и более старшие версии) или AMD EPYC второго поколения или новее (AMD EPYC 7502P и более старшие версии);

- от 8 ядер.

ОЗУ:

- от 8 Гб.

Дисковое пространство:

- от 5 Гб.

#### **2.2.4.1.4. Конфигурация БД Центра**

Процессор:

- от 16 ядер.

ОЗУ:

- от 32 Гб.

Дисковое пространство:

- от 200 Гб.

#### **2.2.4.1.5. Конфигурация БД кэша Центра**

Процессор:

- от 4 ядер.

ОЗУ:

- от 16 Гб.

Дисковое пространство:

- от 5 Гб.

#### **2.2.4.2. Требования к сетевой инфраструктуре**

Пропускная способность канала между серверами приложений и внешними информационными системами, направляющими запросы в ПО DQF: Federation, должна составлять не менее 1 Гбит/с.

Пропускная способность канала между серверами приложений и брокером сообщений должна составлять не менее 1 Гбит/с.

#### **2.2.4.3. Назначение прав для пользователя, под которым будет проводиться установка и настройка ПО DQF: Federation**

Для установки и настройки ПО DQF: Federation не требуется назначение прав пользователя, установку и настройку осуществляет системный администратор, имеющий доступ к серверу приложений, где будет установлено ПО DQF: Federation.

#### **2.2.4.4. Установка и настройка программного окружения**

После подготовки сетевой инфраструктуры и аппаратной платформы необходимо произвести работы по подготовке программного окружения для установки ПО DQF: Federation.

Для установки дистрибутива ПО DQF: Federation необходимо установить и предварительно настроить ПО Docker версии 20.10+ в соответствии с используемой операционной системой. Установку необходимо проводить, следуя инструкциям разработчика программного обеспечения. Инструкция для различных операционных систем доступна по ссылке: <https://docs.docker.com/engine/install/>.

В случае отсутствия в составе программного окружения брокера очередей и/или СУБД, необходимо произвести установку, следуя инструкциям разработчика соответствующего программного обеспечения, доступных по ссылкам:

- ссылка на инструкцию по установке Kafka:

<https://kafka.apache.org/documentation/#quickstart>

– ссылки на дистрибутивы и инструкции по установке СУБД Postgres на различных операционных системах: <https://www.postgresql.org/download/>

– ссылка на инструкцию по установке Redis: [https://redis.io/docs/latest/operate/oss\\_and\\_stack/install/install-redis/](https://redis.io/docs/latest/operate/oss_and_stack/install/install-redis/)

## 2.2.5. Установка и настройка ПО DQF: Federation

### 2.2.5.1. Установка и запуск дистрибутивов ПО DQF: Federation

Для установки дистрибутивов ПО DQF: Federation предварительно необходимо скачать архивы с предоставленными дистрибутивами в формате .tar, а также файлы с шаблонами настроек, необходимых для запуска ПО, и скопировать их на сервера приложений, на которых должно быть развернуто ПО.

#### 2.2.5.1.1. Установка и запуск сервиса Узла

Для установки сервиса Узла необходимо перейти в каталог со скаченным архивом и добавить дистрибутив ПО DQF: Federation (сервис Узел) в локальный Docker Registry, выполнив команду:

```
docker load -i node.tar
```

После чего, используя шаблон настроек, произвести настройку параметров запуска, необходимых для запуска сервиса Узла и специфичных для данного окружения, и сохранить данные настройки в файл env.file:

```
SPRING_KAFKA_BOOTSTRAP_SERVERS: "kafka:9094" # Адрес и порт брокера
Kafka в формате ХОСТ:ПОРТ
SPRING_DATASOURCE_URL: jdbc:postgresql://postgres:5432/node-0 # URL для
подключения к БД Узла
SPRING_DATASOURCE_USERNAME: node # Логин пользователя БД
SPRING_DATASOURCE_PASSWORD: node # Пароль пользователя БД
FMDM_NODE_CENTER_LOAD_TOPIC: load-center # Топик Kafka для передачи
обновлений от сервиса Узла к сервису Центра
FMDM_NODE_NODE_LOAD_TOPIC: load-node-0 # Топик для передачи сообщений в
сервис Узла
LOGGING_LEVEL_RU_CLEANDATA: WARN # Уровень логирования сервиса
```

Затем необходимо создать на сервере приложения директорию /etc/fmdm/ и загрузить в нее файл лицензии полученной от поставщика ПО. После чего запустить сервис следующей командой:

```
docker run -d --env-file ./env.file --name node-0 --mount
'source=/etc/fmdm/,target=/etc/fmdm/' node
```

### 2.2.5.1.2. Установка и запуск сервиса Центра

Для установки сервиса Центра необходимо перейти в каталог со скаченным архивом и добавить дистрибутив ПО DQF: Federation (сервис Центр) в локальный Docker Registry, выполнив команду:

```
docker load -i center.tar
```

После чего, используя шаблон настроек, произвести настройку параметров запуска, необходимых для запуска сервиса Центра и специфичных для данного окружения, и сохранить данные настройки в файл env.file:

```
LOGGING_LEVEL_RU_CLEANDATA: WARN
SPRING_KAFKA_BOOTSTRAP_SERVERS: "kafka:9094" # Адрес и порт брокера Kafka
в формате ХОСТ:ПОРТ
SPRING_DATASOURCE_URL: jdbc:postgresql://postgres:5432/center # URL для
подключения к БД Центра
SPRING_DATASOURCE_USERNAME: center # Логин пользователя БД Центра
SPRING_DATASOURCE_PASSWORD: center # Пароль пользователя БД Центра
LOGGING_LEVEL_RU_CLEANDATA_FMDM: "warn" # Уровень логгирования
сервиса
FMDM_CENTER_LOAD_TOPIC: load-center # Топик Kafka для передачи обновлений
от сервиса Узла к сервису Центра
FMDM_NODES_0_LOAD_TOPIC: load-node-0 # Топик для передачи сообщений в
сервис Узла
FMDM_NODES_0_SUPPORTED_FRAGMENTS_N:
"<свойство>/<компонент>/<фрагмент>" # N-ый фрагмент для сопоставления, ожидаемый
от данного узла.
```

Затем создать на сервере приложения директорию /etc/fmdm/ и загрузить в нее файл лицензии полученной от поставщика ПО. После чего запустить сервис следующей командой:

```
docker run -d --env-file ./env.file --name center --mount
'source=/etc/fmdm/,target=/etc/fmdm/' center
```

### 2.2.5.2. Проверка работоспособности ПО DQF: Federation

После завершения установки и настройки ПО DQF: Federation необходимо выполнить проверку работоспособности ПО DQF: Federation. Для этого требуется направить тестовый запрос, используя установленный ранее брокер очередей Kafka и UI-клиент Kafka, например, Kafka UI (<https://github.com/provectus/kafka-ui>). Ниже приведен пример тестового запроса для DQF: Federation, состоящий из одного экземпляра сервиса Узла, и брокера очередей Kafka с описанием последовательности выполнения необходимых действий:

1. Зайти в UI Kafka.
2. Сформировать и отправить тестовый запрос в очередь Узла, указанную в переменной FMDM\_NODE\_LOAD\_TOPIC файла env.file Узла, созданного в рамках п. 2.2.5.1.1 настоящего Руководства по развертыванию и установке.
3. Проверить, что направленный тестовый запрос обработан ПО DQF: Federation и результат записан в БД Центра.

```
TARGET_CARDS as (  
    select  
        ID as CARD_ID  
    from  
        CARD C  
    where 1=1  
        and (C.SOURCE_ID, C.SOURCE_CARD_ID) in (  
            select * from unnest(:sourceIds, :sourceCardIds) as X(SOURCE_ID,  
SOURCE_CARD_ID)  
        )  
        and not C.DELETED  
        and not C.DEPRECATED  
),  
INCLUDING_CONSOLIDATIONS as (  
    select CONSOLIDATION_ID  
    from  
        CONSOLIDATION_CARD CC  
        inner join CONSOLIDATION C  
            on CC.CONSOLIDATION_ID = C.ID  
    where 1=1  
        and CC.CARD_ID in (select CARD_ID from TARGET_CARDS)  
        and C.TOP_LEVEL  
    )  
select  
    CC.CONSOLIDATION_ID as GROUP_ID,  
    array_agg(C.SOURCE_ID) as SOURCE_IDS,  
    array_agg(C.SOURCE_CARD_ID) as SOURCE_CARD_IDS  
from
```

```
CONSOLIDATION_CARD CC
  inner join CARD C
    on CC.CARD_ID = C.ID
where 1=1
  and CC.CONSOLIDATION_ID in (select CONSOLIDATION_ID from
INCLUDING_CONSOLIDATIONS)
  and (1<>1
  or C.ID in (select CARD_ID from TARGET_CARDS)
  or case when array_length(:sourceIdsFilter, 1)>0
    then C.SOURCE_ID in (select unnest(:sourceIdsFilter))
    else true
  end
)
group by
  CC.CONSOLIDATION_ID
```

Описание форматов для формирования тестового запроса и проверки ответа приведено в документе «Руководство системного администратора программного обеспечения Data Quality Framework» в разделе 3.2.5.

### 2.2.6. Перезапуск

Для перезапуска ПО DQF: Federation необходимо на каждом сервере приложений выполнить команду Docker для перезапуска контейнера.

Узел

```
docker restart node-0
```

Центр

```
docker restart center
```