

СПО TRUSTESS Docs

Руководство администратора

АННОТАЦИЯ

Настоящий документ представляет собой руководство администратора специального программного обеспечения TrustESS Docs и содержит сведения о порядке его установки, настройки и администрирования.

Общие термины и сокращения	5
1 Назначение и условия применения Сервиса документов TrustESS Docs	6
1.1 Назначение	6
1.2 Краткое описание функций и взаимодействия	6
1.3 Условия применения	7
1.4 Характеристики программы.....	8
1.4.1 Режим работы.....	8
1.4.2 Средства контроля правильности выполнения и восстановления	8
1.4.3 Ограничения по применению	8
2 Подготовка окружения перед установкой программы.....	9
2.1 Установка и настройка ViPNet TLS Gateway	9
2.2 Установка и настройка Docker Engine и Docker Compose	12
2.3 Установка и настройка СУБД PostgreSQL.....	12
2.4 Установка и настройка СУБД Redis	13
2.5 Создание файлового хранилища Docker Volumes.....	13
2.6 Подключение к RabbitMQ	14
3 Порядок установки и настройки	15
3.1 Предварительная настройка.....	15
3.2 Подготовка конфигурации	15
3.3 Установка и запуск программы.....	15
3.3.1 Общие сведения	15
3.3.2 С использованием командной оболочки bash.....	15
3.3.3 С использованием стандартной консоли или другой командной оболочки (без использования bash-скрипта).....	16
3.4 Перезапуск программы	16
3.4.1 Общие сведения	16
3.4.2 Перезапуск с использованием командной оболочки bash.....	16
3.4.3 Перезапуск с использованием стандартной консоли или другой командной оболочки (без использования bash-скрипта)	16
3.5 Изменение параметров конфигурации	16
3.6 Обновление.....	17
3.6.1 Общие сведения	17
3.6.2 Обновление с использованием командной оболочки bash.....	17
3.6.3 Обновление с использованием стандартной консоли или другой командной оболочки (без использования bash-скрипта)	17
3.7 Масштабирование (HL).....	17
3.7.1 Общие сведения	17
3.7.2 Настройка PostgreSQL при масштабировании	18
3.7.3 Настройка RabbitMQ при масштабировании	18
3.7.4 Настройка Redis при масштабировании	18
3.7.5 Настройка Файлового хранилища при масштабировании.....	18
3.8 Создание, редактирование и удаление пользователей.....	18
3.9 Логирование.....	18
3.10 Мониторинг сервиса.....	19
3.11 Рекомендации по мониторингу метрик	21
3.11.1 Общие сведения	21
3.11.2 Оптимальный набор метрик для мониторинга	21

3.12 Резервное копирование и восстановление (НА).....	22
4 Возможные неполадки и способы их устранения.....	23
4.1 Общие сведения	23
4.2 Потеря связности контейнеров.....	23
4.2.1 Описание ошибки:.....	23
4.2.2 Внешнее проявление (симптомы).....	23
4.2.3 Решение.....	23
4.3 Нехватка количества соединений с БД	23
4.3.1 Описание	23
4.3.2 Внешнее проявление (симптомы).....	23
4.3.3 Решение.....	24
Приложение 1. Параметры конфигурации, задаваемые в конфигурационном файле docs.env	25

ЕПГУ	Единый портал государственных и муниципальных услуг
ЗАС	Защищенная автоматизированная система, размещаемая в инфраструктуре Потребителя
Заказчик	Юридическое лицо, которое предоставляет Клиентам услуги с использованием Сервиса документов
Идентификация	Процесс подтверждения личности Клиента.
МП	Мобильное приложение
Пользователь/Клиент	Лицо, выступающее в роли владельца сертификата ключа проверки электронной подписи, используемого для формирования электронной подписи с использованием МП и СЗАС
Потребитель	Организация, являющаяся потребителем сервисов СЗАС и использующая в своих бизнес-процессах возможность подписания электронной подписью пользователей формируемых организацией документов
СЗАС	Специализированная защищенная автоматизированная система, соответствующая требованиям, указанным в п. 14 и 15 документа "Правила создания и использования сертификата ключа проверки усиленной неквалифицированной электронной подписи в инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг в электронной форме", утвержденного постановлением Правительства Российской Федерации от 1 декабря 2021 г. № 2152
Сервис документов	Сервис документооборота, развертываемый в ЗАС Потребителя с использованием СПО TrustESS Docs
СПО TrustESS Docs	Специальное программное обеспечение TrustESS Docs, обеспечивающее взаимодействие и документооборот между мобильным приложением пользователей и ЗАС Потребителя
СД	Сервис документов
СПО	Специальное программное обеспечение
УКЭП	Усиленная квалифицированная электронная подпись
УНЭП	Усиленная неквалифицированная электронная подпись
УКЭП дл	УКЭП государственного органа или должностного лица государственного органа, в соответствии с 63-ФЗ: ст. 17.2, ч.3; ст. 17.4

1.1 НАЗНАЧЕНИЕ

Специальное программное обеспечение TrustESS Docs (далее – СПО TrustESS Docs) предназначено для развертывания Сервиса документов (далее – СД) с целью получения, хранения и передачи файлов на подпись в МП пользователя, а также инициации потоков работ на подпись в Сервис авторизации со стороны заказчика.

1.2 КРАТКОЕ ОПИСАНИЕ ФУНКЦИЙ И ВЗАИМОДЕЙСТВИЯ

СПО TrustESS Docs обеспечивает взаимодействие с мобильным приложением пользователей с целью выполнения процесса отправки документов на подписание в МП и получения подписи после подписания.

СПО TrustESS Docs предназначено для установки в защищенной автоматизированной системе Потребителя (далее – ЗАС) и обеспечивает выполнение основной функции СД по предоставлению API для создания и получения информации о потоках работ на подпись документов (в том числе файлов подписи) для Заказчика:

1. Запрос на авторизацию пользователей.
2. Запрос на создание потока.
3. Запрос на получение информации о потоке.
4. Запрос на скачивание файла документа и подписи к нему.

Файлы, полученные от Заказчика через СД, хранятся только на стороне СД и не передаются в Сервис авторизации (далее – СА).

В ходе функционирования СД обеспечивает:

1. Взаимодействие СД с мобильным приложением пользователя с использованием алгоритмов двустороннего ГОСТ TLS.
2. Взаимодействие ЗАС Заказчика с СД по открытому протоколу HTTP с использованием токена авторизации.
3. Взаимодействие СД с СА по открытому протоколу HTTP внутри защищенного канала по классу KB.
4. Хранение файлов в NAS.

Упрощенная схема взаимодействия СД показана на рисунке 1.1.

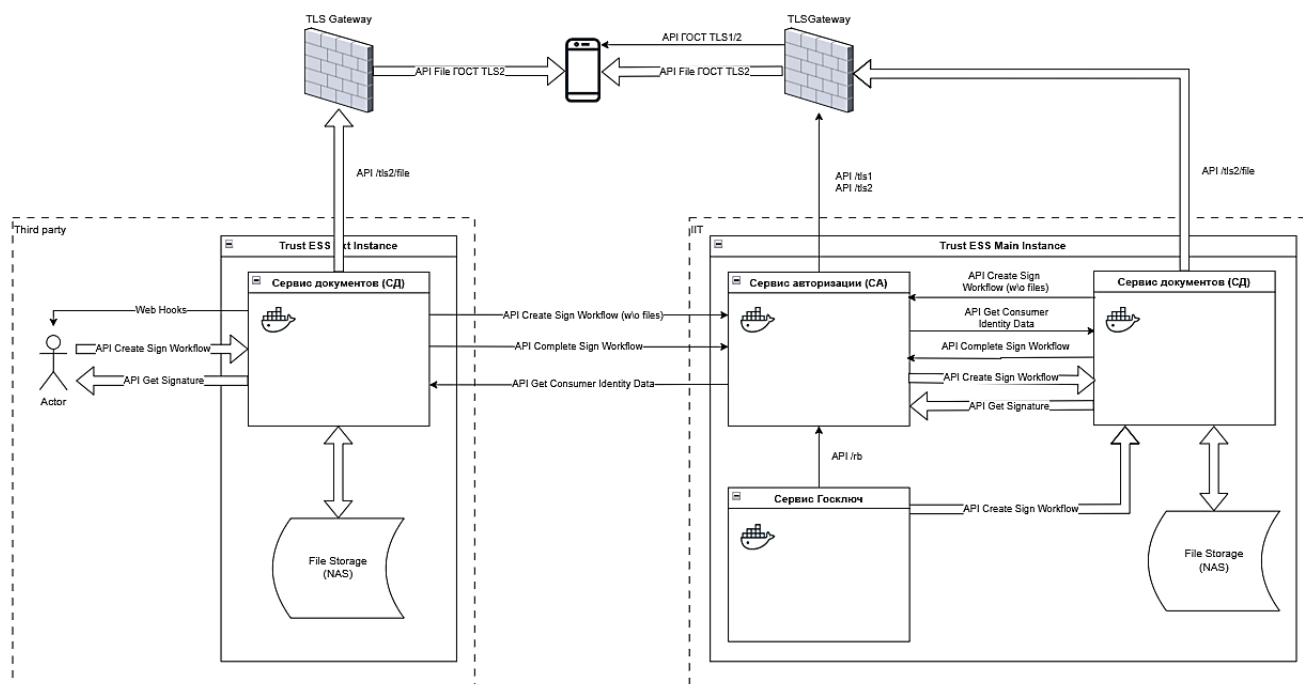


Рисунок 1.1 – Упрощенная схема взаимодействия СД с внешними системами

1.3 УСЛОВИЯ ПРИМЕНЕНИЯ

СПО TrustESS Docs предназначено для функционирования в среде операционной системы Astra Linux Special Edition 1.7 (в режиме замкнутой программной среды), имеющей действующий сертификат соответствия ФСБ России № СФ/СЗИ-0614 от 29 марта 2023 года и эксплуатации на серверном оборудовании, поддерживающем архитектуру x86, x86-64 с минимальной рекомендуемой производителем операционной системы (далее – ОС) аппаратной конфигурацией.

На оборудовании должно быть установлено дополнительное программное обеспечение, указанное в таблице 1.1.

Таблица 1.1 – Перечень требуемого дополнительного программного обеспечения

Название	Версия	Назначение
ViPNet TLS Gateway	2.2.2 и выше	Защита канала обмена данными между МП пользователя и СД, путем обеспечения защищенного удаленного подключения.
Redis	5.0 и выше	Нереляционная СУБД, предназначенная для хранения кэша и результатов выполнения запросов.
PostgreSQL	16 и выше	Реляционная СУБД, предназначенная для постоянного хранения: – данных потоков на подпись документов; – сведений о файлах документов; – дополнительных служебных данных. <i>Примечания:</i> 1. Для базы данных возможно использование существующего экземпляра PostgreSQL или создание отдельного экземпляра специально для СД. 2. Вариант исполнения PostgreSQL не имеет значения, важна только его версия.
docker compose	2.29.6 и выше	Предназначен для использования при запуске контейнеров.
docker engine	24.0.2 и выше	Клиент-серверное приложение платформы Docker.
docker volume	Последняя стабильная версия	Предназначен для использования в качестве файлового хранилища.
RabbitMQ	Последняя стабильная версия	Брокер сообщений <i>Примечание.</i> В СПО TrustESS Docs используется стандартный протокол обмена сообщениями AMQP 0.9.1, поэтому рекомендуется использовать последнюю версию RabbitMQ, поддерживающую данный интерфейс и имеющую минимальное количество уязвимостей.

Установка, предварительная настройка и эксплуатация СПО TrustESS Docs осуществляется в соответствии с эксплуатационной документацией, входящей в комплект поставки.

СПО TrustESS Docs может быть установлено на виртуальные машины.

При установке СПО TrustESS Docs рекомендуется дополнительно руководствоваться документацией на ОС.

Для сохранения бинарной целостности запрещается устанавливать обновления сертифицированного программного изделия.

Перед началом эксплуатации СПО TrustESS Docs необходимо установить все доступные обновления дополнительного программного обеспечения, указанного в таблице 1.1.

ВНИМАНИЕ!

СПО TrustESS Docs устанавливается в docker-контейнерах, поэтому обновление docker engine, установленного на хостовой ОС, необходимо выполнять до начала установки СПО TrustESS Docs, так как обновление docker engine в ходе эксплуатации может привести к нарушению его функционирования. Обновление ОС и других ее компонентов не влияет на функционирование СПО TrustESS Docs.

1.4 ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.4.1 РЕЖИМ РАБОТЫ

Режим работы СПО TrustESS Docs – круглосуточный (24/7).

1.4.2 СРЕДСТВА КОНТРОЛЯ ПРАВИЛЬНОСТИ ВЫПОЛНЕНИЯ И ВОССТАНОВЛЕНИЯ

Контроль правильности выполнения осуществляется средствами мониторинга (см. п. **3.10 Мониторинг сервиса**).

Восстановление работоспособности СПО TrustESS Docs после сбоя внешних сервисов осуществляется автоматически после восстановления их работоспособности и соединения с ними.

1.4.3 ОГРАНИЧЕНИЯ ПО ПРИМЕНЕНИЮ

При функционировании СД должны быть соблюдены следующие условия:

1. Пользователь установил и зарегистрировал МП.
2. Для пользователя выпущен сертификат ЭП.

2 ПОДГОТОВКА ОКРУЖЕНИЯ ПЕРЕД УСТАНОВКОЙ ПРОГРАММЫ

2.1 УСТАНОВКА И НАСТРОЙКА VIPNET TLS GATEWAY

Установка и базовая настройка ViPNet TLS Gateway выполняется в соответствии с документацией на изделие.

Примечание. Комплект документации на ViPNet TLS Gateway можно скачать на официальном сайте разработчика: <https://infotecs.ru/products/vipnet-tls-gateway/?ysclid=m905gh2npx74884296>.

После установки и базовой настройки ViPNet TLS Gateway выполните его окончательную настройку для работы в СД:

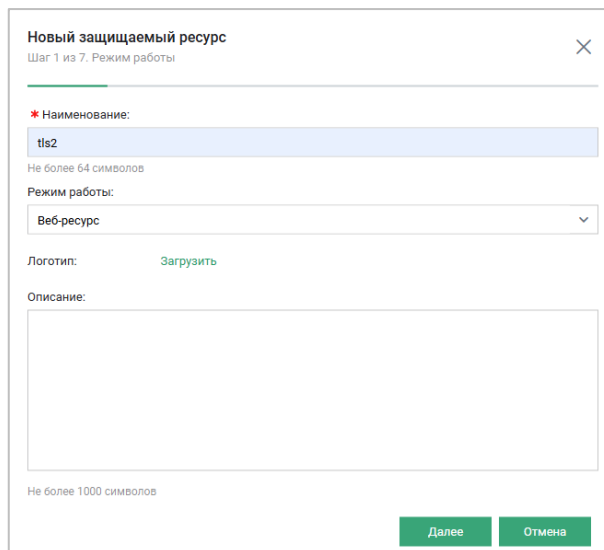
1. Добавьте защищаемый двусторонним TLS ГОСТ веб-ресурс.

Примечание. Порядок добавления ресурса указан в разделе «Управление параметрами доступа к ресурсам» Руководства администратора на ViPNet TLS Gateway (ФРКЕ.00169-0232 01).

2. Выполните настройку защищаемого веб-ресурса:

2.1. Выполните Шаг 1 (Рисунок 2.1):

- укажите наименование ресурса – **tls2**;
- задайте режим работы – **Веб-ресурс**;
- нажмите **Далее**.



Новый защищаемый ресурс
Шаг 1 из 7. Режим работы

* Наименование:
tls2
Не более 64 символов

Режим работы:
Веб-ресурс

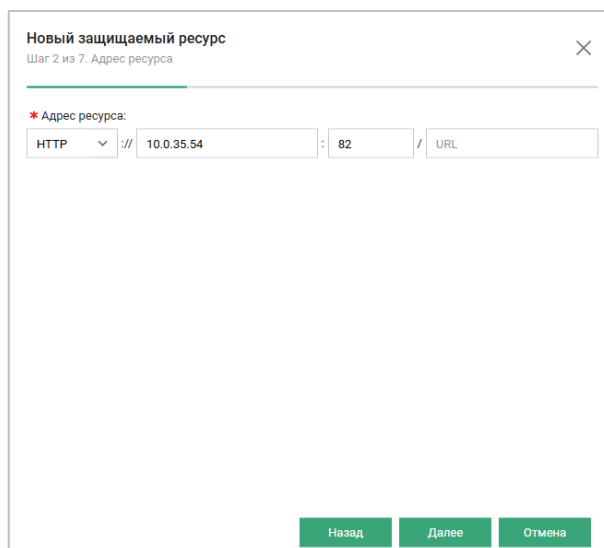
Логотип: Загрузить

Описание:
Не более 1000 символов

Далее Отмена

Рисунок 2.1 – Шаг 1 настройки защищаемого ресурса

- 2.2. На шаге 2 укажите адрес ресурса (формат адреса: **http://<доступный адрес и порт TrustESS Docs>** и нажмите **Далее** (Рисунок 2.2).



Новый защищаемый ресурс
Шаг 2 из 7. Адрес ресурса

* Адрес ресурса:
HTTP 10.0.35.54 : 82 / URL

Назад Далее Отмена

Рисунок 2.2 – Шаг 2 настройки защищаемого ресурса

2.3. Выполните **Шаг 3** (Рисунок 2.3):

- Укажите параметры аутентификации:
 - **Двусторонняя;**
 - **Без проверки прав доступа к ресурсу.**
- Установите Алгоритм подключения:
 - **ГОСТ Р 34.10-2012.**
- Нажмите **Далее**.

Новый защищаемый ресурс
Шаг 3 из 7. Безопасность

Аутентификация

- ☒ Двусторонняя
- ☐ Односторонняя

☒ Без проверки прав доступа к ресурсу ?

☐ Поддержка NTLM-аутентификации

Алгоритмы подключения ?

- ☒ ГОСТ Р 34.10-2012
- ☐ RSA
- ☐ ECDSA

Назад Далее Отмена

Рисунок 2.3 – Шаг 3 настройки защищаемого ресурса

2.4. На **Шаге 4** укажите Адрес прослушивания (Рисунок 2.4):

- Установите флажок в чекбоксе «**Использовать выделенный порт прослушивания для данного ресурса**»;
- Укажите порт (порт любой, на который потом возможно настроить NAT);
- Нажмите **Далее**.

Новый защищаемый ресурс
Шаг 4 из 7. Адреса прослушивания

☒ Использовать выделенный порт прослушивания для этого ресурса 40002

Назначенные системой адреса прослушивания ?

IP-адрес (интерфейс)	Порт
10.0.35.40 (enp11s0)	40002

Дополнительные адреса прослушивания

+ Добавить

Назад Далее Отмена

Рисунок 2.4 – Шаг 4 настройки защищаемого ресурса

2.5. Выполните **Шаг 5** (см. Рисунок 2.5):

- укажите **Суффикс: tls2**
- Установите флажок в чекбоксе **Адрес ресурса на TLS Gateway**.

Новый защищаемый ресурс
Шаг 5 из 7. Адреса подключения

Адреса подключения, назначенные системой ?

* Суффикс:
tls2

Отсутствуют.

Настраиваемые адреса подключения [Изменить](#)

Использовать для подключения:

☒ Адрес ресурса на TLS Gateway
☐ Заданное доменное имя

Адреса подключения	Использование
https://10.0.35.40:30000	Да
https://193.162.30.55:30000	Да
https://trust-entry-reg-demo...:30000	Да
https://trust-entry-reg-test...:30000	Да
https://trust-entry-tls-rc.i...:30000	Да

[Назад](#) [Далее](#) [Отмена](#)

Рисунок 2.5 – Шаг 5 настройки защищаемого ресурса

- 2.6. На **Шаге 6** (Рисунок 2.6) установите флажок в чекбоксе «**Использовать HTTP-заголовки безопасности**» и нажмите **Далее**.

Новый защищаемый ресурс
Шаг 6 из 7. HTTP-заголовки

По умолчанию

В запросы к защищаемым ресурсам всегда включаются HTTP-заголовки X-Real-IP и X-Forwarded-For. Отключить их нельзя.

HTTP-заголовки безопасности

☒ Использовать HTTP-заголовки безопасности ?

Дополнительные HTTP-заголовки

☐ Передавать данные в HTTP-заголовках запроса ?

[Назад](#) [Далее](#) [Отмена](#)

Рисунок 2.6 – Шаг 6 настройки защищаемого ресурса

- 2.7. Выполните **Шаг 7** (Рисунок 2.7):

- Проверьте установленные параметры;
- Нажмите **Сохранить**.

Новый защищаемый ресурс
Шаг 7 из 7. Проверка данных

Наименование:	tls2
Режим работы:	Веб-ресурс
Адрес ресурса:	http://10.0.35.54:82
Аутентификация:	Двусторонняя
Алгоритмы подключения:	ГОСТ Р 34.10-2012
Выделенный порт для прослушивания ресурса:	30000
Суффикс:	tls3
Адреса подключения:	Адрес ресурса на TLS Gateway
Использование HTTP-заголовков безопасности:	Включено

[Назад](#) [Сохранить](#) [Отмена](#)

Рисунок 2.7 – Шаг 7 настройки защищаемого ресурса

3. Добавьте в доверенные корневые сертификаты УЦ с указанием актуальных CDP для автоматического обновления CRL.
4. Создать и передать запрос в ИИТ, а затем установить транспортный сертификат, выпущенный доверенным УЦ в соответствии с разделом «Работа с транспортными сертификатами» Руководства администратора на ViPNet TLS Gateway (ФРКЕ.00169-0232 01).
5. Завершите настройку ViPNet TLS Gateway:
 - 5.1. Настройте TLS-подключение (**Рисунок 2.8**):
 - для доступа администраторов укажите версии 1.2 и 1.3 протокола TLS;
 - для доступа пользователями укажите версию 1.2 протокола TLS.

Рисунок 2.8 – Настройка TLS-подключения

Примечание. Настройка TLS-подключения осуществляется с целью исключения использования запрещенных протоколов TLS, и осуществляется в соответствии с требованиями разделов «Изменение версии протокола TLS для подключения администраторов» и «Доступ пользователей» Руководства администратора на ViPNet TLS Gateway (ФРКЕ.00169-0232 01).

- 5.2. Задайте максимальный размер тела запроса пользователя не менее 100 Мб (**Рисунок 2.9**).

Рисунок 2.9 – Настройка максимального размера тела запроса пользователя

Примечание. Настройка осуществляется в соответствии с требованиями раздела «Дополнительные настройки параметров подключения к ViPNet TLS Gateway» Руководства администратора на ViPNet TLS Gateway (ФРКЕ.00169-0232 01).

2.2 УСТАНОВКА И НАСТРОЙКА DOCKER ENGINE И DOCKER COMPOSE

Дистрибутив СПО TrustESS Docs представляет собой образ контейнера в формате Docker.

Для его успешного запуска СПО TrustESS Docs требуется наличие установленного программного обеспечения Docker Engine и Docker Compose на сервере.

Установка программного обеспечения Docker Engine и Docker Compose осуществляется из репозитория ОС.

Для установки программного обеспечения Docker Engine и Docker Compose используется встроенный пакетный менеджер ОС, в котором необходимо выполнить команды:

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli docker-compose-plugin
```

2.3 УСТАНОВКА И НАСТРОЙКА СУБД POSTGRESQL

Для работы СД возможно использование БД в составе существующего экземпляра PostgreSQL или отдельного экземпляра PostgreSQL, развернутого специально для сервиса.

Для развертывания отдельного экземпляра PostgreSQL в Docker используется немодифицированный образ контейнера с PostgreSQL версии 16:

```
docker pull postgres:16
```

Для запуска Docker-контейнера выполните команду:

```
docker run -d postgres:16
```

Для функционирования СД требуется выполнить настройку подключения СУБД PostgreSQL.

Настройка подключения осуществляется через файл docs.env путем задания значений следующих переменных окружения Docker-контейнера:

- DATABASE_HOST – имя хоста (по умолчанию 127.0.0.1)
- DATABASE_PORT – порт (по умолчанию 5432)
- DATABASE_USER – имя пользователя (по умолчанию docs)
- DATABASE_PASSWORD – пароль пользователя (по умолчанию 11111111)
- DATABASE_NAME – имя базы данных (по умолчанию docs)
- DATABASE_POOL_SIZE – размер пула подключений на каждый запущенный Docker-контейнер (по умолчанию 5)

2.4 УСТАНОВКА И НАСТРОЙКА СУБД REDIS

Для развертывания СУБД Redis в Docker используется немодифицированный Docker-образ разработчика.

Для функционирования СД требуется выполнить настройку подключения СУБД Redis.

Настройка подключения осуществляется через файл docs.env путем задания значений следующих переменных окружения Docker-контейнера:

- REDIS_URL – строка подключения к Redis (по умолчанию redis://redis:6379/1).

2.5 СОЗДАНИЕ ФАЙЛОВОГО ХРАНИЛИЩА DOCKER VOLUMES

В качестве файлового хранилища используется Docker Volumes, подключенный к Docker-контейнерам с СПО TrustESS Docs.

*Примечание. Монтирование docker volumes должно быть выполнено ко всем Docker-контейнерам с инфиксом *-docs*.*

Docker Volumes может быть создан с указанием любого поддерживаемого Docker драйвера (<https://docs.docker.com/engine/storage/volumes/#use-a-volume-driver>).

Docker Volumes должен быть создан до первого запуска Docker-контейнера с сервисом и не удаляться на всем протяжении пользования сервисом.

❗ ВНИМАНИЕ!

Удаление файлового хранилища Docker Volumes приведет к потере хранящихся на нем файлов

Файловое хранилище Docker Volumes должно иметь имя "trust_entry_docs_storage".

Пример создания Docker Volumes с хранением данных на внешнем NAS с NFSv4:

```
docker volume create \
--driver local \
--opt type=nfs \
--opt o=addr=x.x.x.x,nfsvers=4 \
--opt device=/mnt/data/docs \
trust_entry_docs_storage
```

Примечание. При первом запуске СПО TrustESS Docs с помощью Docker Compose, созданный Docker Volumes будет подключен ко всем необходимым контейнерам.

Для определения дискового пространства, занимаемого Docker Volumes, используйте команду:

```
docker system df -v
```

Примечание. Определение дискового пространства, занимаемого Docker Volumes, может занять продолжительное время.

Способы резервного копирования файлов, хранящихся в Docker Volumes, описаны в пункте **3.12 Резервное копирование и восстановление (HA)**.

2.6 ПОДКЛЮЧЕНИЕ К RABBITMQ

Для функционирования СД требуется сетевое подключение к AMQP-брокеру (RabbitMQ).

Подключение осуществляется стандартными методами и не требует специальных настроек как самого сервиса, так и RabbitMQ.

Параметры, содержащиеся в конфигурационном файле и используемые при настройке подключения, указаны в **приложении 1**.

3 ПОРЯДОК УСТАНОВКИ И НАСТРОЙКИ

3.1 ПРЕДВАРИТЕЛЬНАЯ НАСТРОЙКА

Перед установкой СПО TrustESS Docs требуется выполнить предварительную проверку и настройку окружения:

1. Убедиться в работоспособности всех компонент окружения:
 - PostgreSQL;
 - Redis;
 - RabbitMQ;
 - Файлового хранилища.
2. Создать необходимые учетные записи пользователей в компонентах окружения (при использовании локальных компонентов выполняется автоматически скриптом при первой установке).
3. Создать файл конфигурации СД с необходимым набором параметров (см. п. **3.2 Подготовка конфигурации**)
4. Разместить файлы конфигурации СД в директорию **./config**,

*Примечание. Директория **./config**, в которую помещаются файлы конфигурации, должна находиться в директории запуска СПО TrustESS Docs.*

5. Скопировать архив СПО TrustESS Docs, содержащий Docker-образ контейнера и скрипты установки, в директорию, из которой будет осуществляться запуск сервиса.
6. Убедиться в доступности внешнего порта сервиса – 15262 (порт должен быть свободен).

3.2 ПОДГОТОВКА КОНФИГУРАЦИИ

Перед установкой необходимо подготовить конфигурацию сервиса.

Файл конфигурации располагается в директории **./config/env/docs.env** (путь относительно директории, в которой находится файл **docker-compose.docs.yml**).

Изменяемые параметры конфигурации файла **docs.env**, и их значения, заданные по умолчанию, указаны в приложении 1.

Примечания:

1. Если параметр не указан в файле конфигурации, то значит, что используется значение по умолчанию.
2. Если в файле конфигурации указан параметр с пустым значением, то используется пустое значение.
3. При изменении значения параметров конфигурации уже после установки сервиса требуется пересоздание Docker-контейнеров с сервисами (см. п. **3.3 Установка и запуск программы**).

3.3 УСТАНОВКА И ЗАПУСК ПРОГРАММЫ

3.3.1 ОБЩИЕ СВЕДЕНИЯ

Установка и запуск СПО TrustESS Docs может осуществляться двумя способами:

1. С использованием bash-скрипта, запускаемого в командной оболочке **bash**;
2. С использованием стандартной консоли ОС.

3.3.2 С ИСПОЛЬЗОВАНИЕМ КОМАНДНОЙ ОБОЛОЧКИ BASH

При использовании командной оболочки **bash** установка и запуск СПО TrustESS Docs автоматизированы с использованием bash-скрипта: **manage.sh**.

Для первоначальной установки СПО TrustESS Docs необходимо в командной оболочке **bash** запустить скрипт установки выполнив команду:

```
./manage.sh install
```

В процессе установки будет выведена информация о процессе установки и запрошены учетные данные для учетной записи Администратора.

3.3.3 С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНОЙ КОНСОЛИ ИЛИ ДРУГОЙ КОМАНДНОЙ ОБОЛОЧКИ (БЕЗ ИСПОЛЬЗОВАНИЯ BASH-СКРИПТА)

При использовании стандартной консоли или другой командной оболочки, отличной от bash, установка и запуск СПО TrustESS Docs осуществляется путем последовательного выполнения команд.

Для установки и запуск СПО TrustESS Docs необходимо:

1. Загрузить образ Docker-контейнера в хранилище Docker командой:

```
docker load < docs.tar.gz
```

2. Создать и запустить Docker-контейнеры с сервисом командой:

```
docker compose -f docker-compose.docs.yml --env-file docs.compose.env -p trust_entry up -d
```

3. Создать УЗ Администратора выполнив команду:

```
docker exec -it trust_entry-docs_backend-1 python manage.py createadmin
```

Примечания:

1. При создании УЗ Администратора следуйте командам, выводимым в консоли.
2. Для того, чтобы иметь возможность запускать команды от непривилегированного linux-пользователя без sudo, необходимо добавить этого пользователя в группу docker (`usermod -aG docker <username>`).

После успешного запуска всех контейнеров для проверки работоспособности сервиса необходимо:

1. Открыть Панель администрирования по адресу `http://адрес сервера:8080/`;
2. Войти под УЗ Администратор с учетными данными, созданными выше.

3.4 ПЕРЕЗАПУСК ПРОГРАММЫ

3.4.1 ОБЩИЕ СВЕДЕНИЯ

Перезапуск СПО TrustESS Docs аналогично установке и запуску, может осуществляться двумя способами:

1. С использованием bash-скрипта, запускаемого в командной оболочке bash;
2. С использованием стандартной консоли ОС.

3.4.2 ПЕРЕЗАПУСК С ИСПОЛЬЗОВАНИЕМ КОМАНДНОЙ ОБОЛОЧКИ BASH

Для перезапуска Docker-контейнеров СПО TrustESS Docs с использованием командной оболочки bash необходимо запустить bash-скрипт: `manage.sh` выполнив команду:

```
./manage.sh restart
```

3.4.3 ПЕРЕЗАПУСК С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНОЙ КОНСОЛИ ИЛИ ДРУГОЙ КОМАНДНОЙ ОБОЛОЧКИ (БЕЗ ИСПОЛЬЗОВАНИЯ BASH-СКРИПТА)

Для перезапуска Docker-контейнеров СПО TrustESS Docs с использованием стандартной консоли или командной оболочки (без использования bash-скрипта), необходимо в консоли выполнить команду:

```
docker compose -f docker-compose.docs.yml --env-file docs.compose.env -p trust_entry restart
```

3.5 ИЗМЕНЕНИЕ ПАРАМЕТРОВ КОНФИГУРАЦИИ

Для изменения параметров конфигурации СД

1. Перейдите в директорию, в которой хранится файл конфигурации (см. п. 3.2 Подготовка конфигурации);
2. Внесите изменения параметров конфигурации в файле `docs.env`;
3. Выполнить пересоздание Docker-контейнеров командой (полная переустановка не требуется):

```
docker compose -f docker-compose.docs.yml --env-file docs.compose.env -p trust_entry up -d
```

3.6 ОБНОВЛЕНИЕ

3.6.1 ОБЩИЕ СВЕДЕНИЯ

Обновление СПО TrustESS Docs может осуществляться двумя способами:

1. С использованием bash-скрипта, запускаемого в командной оболочке bash;
2. С использованием стандартной консоли ОС.

ВНИМАНИЕ!

В процессе обновления сервис недоступен.

Перед началом обновления требуется скачать и поместить в одну директорию со скриптом manage.sh:

- архив docs.tar.gz, содержащий новую версию Docker-образа контейнера с СПО TrustESS Docs;
- конфигурационные файлы docs.compose.env и docker-compose.docs.yml.

3.6.2 ОБНОВЛЕНИЕ С ИСПОЛЬЗОВАНИЕМ КОМАНДНОЙ ОБОЛОЧКИ BASH

Для перезапуска Docker-контейнеров СПО TrustESS Docs с использованием командной оболочки bash необходимо запустить bash-скрипт: manage.sh выполнив команду:

```
./manage.sh update
```

Дождаться завершения обновления.

3.6.3 ОБНОВЛЕНИЕ С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНОЙ КОНСОЛИ ИЛИ ДРУГОЙ КОМАНДНОЙ ОБОЛОЧКИ (БЕЗ ИСПОЛЬЗОВАНИЯ BASH-СКРИПТА)

Для обновления Docker-контейнеров СПО TrustESS Docs с использованием стандартной консоли или командной оболочки (без использования bash-скрипта), необходимо:

1. Загрузить образ Docker-контейнера в хранилище Docker командой:

```
docker load < docs.tar.gz
```

2. Остановить работающие Docker-контейнеры с сервисом командой:

```
docker compose -f docker-compose.docs.yml --env-file docs.compose.env -p trust_entry down
```

3. Запустить Docker-контейнеры с сервисом командой:

```
docker compose -f docker-compose.docs.yml --env-file docs.compose.env -p trust_entry up -d
```

3.7 МАСШТАБИРОВАНИЕ (HL)

3.7.1 ОБЩИЕ СВЕДЕНИЯ

Для масштабирования сервиса:

1. Все компоненты должны иметь возможность доступа на чтение и запись (MASTER-MASTER репликация);
2. Должно быть выполнено отдельное масштабирование каждого компонента и обеспечен общий доступ к кеш, очередям, базе данных и файловому хранилищу.

Сервис не хранит состояние внутри и не взаимодействует с другими своими Docker-контейнерами напрямую через Docker-сеть, поэтому масштабирование сервиса осуществляется путем развертывания дополнительных экземпляров Docker-контейнеров.

Развертывание дополнительных экземпляров Docker-контейнеров может быть выполнено на одном или нескольких (разных) серверах.

Для доступа к сервису используется TCP-порт: 15262. Отправка запросов только на данный порт требует балансировки, которая обеспечивается добавлением компонента балансировки HTTP-запросов (Nginx, HAProxy) на запущенные экземпляры сервиса

Другие компоненты, входящие в состав сервиса и хранящие состояние, должны быть масштабированы в соответствии с документацией на данные компоненты или с использованием готовых решений, имеющих стандартный для каждого из них способ подключения и протокол взаимодействия.

3.7.2 НАСТРОЙКА POSTGRESQL ПРИ МАСШТАБИРОВАНИИ

Сервис использует одну БД "docs", в которой могут содержаться таблицы, ключи, индексы и последовательности (не используются хранимые процедуры, триггеры).

При масштабировании рекомендуется использовать решения с **master-master** репликацией (например: <https://postgrespro.ru/docs/enterprise/current/multimaster>). В случае отсутствия данного решения, возможно использование стандартной стриминговой **master-slave** репликации с вынесением на реплику части читающей нагрузки.

3.7.3 НАСТРОЙКА RABBITMQ ПРИ МАСШТАБИРОВАНИИ

Сервис использует классические очереди в virtual host "docs".

RabbitMQ обеспечивает поддержку кластеризации «из коробки» (см. <https://www.rabbitmq.com/docs/clustering>).

Настройка подключения RabbitMQ осуществляется с помощью переменных окружения Docker-контейнера.

3.7.4 НАСТРОЙКА REDIS ПРИ МАСШТАБИРОВАНИИ

Redis обеспечивает поддержку кластеризации «из коробки» (см. <https://redis.io/topics/cluster-tutorial>).

Настройка подключения Redis осуществляется с помощью переменных окружения Docker-контейнера. По умолчанию используется БД с индексом 1.

3.7.5 НАСТРОЙКА ФАЙЛОВОГО ХРАНИЛИЩА ПРИ МАСШТАБИРОВАНИИ

Хранение файлов Сервисом осуществляется с использованием локальной директории **/app/static**, на которую смонтирован docker volume.

Для docker volume может использоваться любой доступный драйвер (из коробки или сторонний) – <https://docs.docker.com/engine/storage/volumes>.

При настройке файлового хранилища должен быть установлен доступ на чтение и запись.

ВНИМАНИЕ!

В настоящее время удаление файлов не осуществляется!

3.8 СОЗДАНИЕ, РЕДАКТИРОВАНИЕ И УДАЛЕНИЕ ПОЛЬЗОВАТЕЛЕЙ

Работа с пользователями сервиса происходит через web-интерфейс **Панели администрирования**, расположенной по адресу: `http://адрес сервера:8080/home/users`.

Для доступа к **Панели администрирования** используются учетные данные **Администратора**, созданные при установке (см. п. **3.3 Установка и запуск программы**).

3.9 ЛОГИРОВАНИЕ

Логирование во всех контейнерах, связанных с сервисом, происходит методом вывода в консоль (stdout) внутри каждого контейнера.

Для сбора логов возможно использование любого драйвера, поддерживаемого Docker:

<https://docs.docker.com/engine/logging/configure/#supported-logging-drivers>

По умолчанию Docker настроен на сбор логов в JSON файл (через драйвер json-file), вывод их для анализа выполняется командой:

```
docker logs.
```

При использовании json-file, необходимо в обязательном порядке установить ограничение на размер файлов логов и их ротацию в файле конфигурации /etc/docker/daemon.json

Примечание. Порядок использования json-file, настройки ограничения указан в источнике:
<https://docs.docker.com/engine/logging/drivers/json-file/#usage>

В СД используется драйвер gelf (GrayLog, OpenSearch) конфигурируемый у каждого сервиса через docker-compose.yml:

```
logging:
  driver: gelf
  options:
    gelf-address: ${GELF_ADDRESS}
    tag: web
```

где: \${GELF_ADDRESS} -- переменная окружения с адресом и портом расположения GrayLog/OpenSearch, значение которой задается в файле docs.compose.env.

Для применения \${GELF_ADDRESS} после изменения необходимо пересоздать контейнеры используя команду

```
docker(-)compose ... --env-file docs.compose.env up -d
```

Для автоматизации конфигурирования логирования сервисов СД рекомендуется перенаправить все логи docker в выбранный сборщик логов. Это позволит исключить конфигурирование логирования у каждого сервиса.

Для этого необходимо в глобальный файл конфигурации Docker /etc/docker/daemon.json (может отсутствовать и при необходимости его необходимо создать) добавить следующие строки:

```
{
  "log-driver": "gelf",
  "log-opts": {
    "gelf-address": "udp://1.2.3.4:12201"
  }
}
```

После изменения конфигурации логирования необходимо выполнить перезагрузку демона docker командой:

```
$ sudo systemctl restart docker
```

3.10 МОНИТОРИНГ СЕРВИСА

Мониторинг сервиса осуществляется по следующим аспектам:

1. Доступность открытых портов сервиса и его компонентов

Проверяется доступность портов, используемых по умолчанию:

- **PostgreSQL**

TCP порт для мониторинга: 5432

- **RabbitMQ**

TCP порт для мониторинга: 5672

- **Redis**

TCP порт для мониторинга: 6379

- **Docs**

TCP порт для мониторинга: 15262

2. Свободное дисковое пространство

Осуществляется мониторинг свободного дискового пространства в месте размещения сервиса и в месте хранения файлов от сервиса. Каждое из мест требуется мониторить отдельно.

3. Связность компонентов (+healthcheck)

Осуществляется проверка доступности и состояния работы компонентов со стороны сервиса Docs. Для запуска проверки отправляется запрос:

```
GET http://<host>:15262/api/healthcheck
```

где **host** – адрес, на котором развернут сервис Docs.

Пример результата ответа (state = 0 – OK):

```
{
  "datetime": "2025-03-24T06:41:12.525672",
  "version": "dev",
  "state": 0,
  "description": "Микросервис работоспособен",
  "components": [
    {
      "name": "ВД",
      "state": 0,
      "description": "Компонент работоспособен"
    },
    {
      "name": "Брокер сообщений",
      "state": 0,
      "description": "Компонент работоспособен"
    },
    {
      "name": "Файловое хранилище",
      "state": 0,
      "description": "Компонент работоспособен"
    },
    {
      "name": "Redis",
      "state": 0,
      "description": "Компонент работоспособен"
    }
  ]
}
```

Примечание. Интерпретация результата проверки осуществляется по значению параметра *state* в ответе:

- если *state* = 0, то проверка пройдена успешно;
- если *state* = 1, то обнаружена недоступность или сбой в работе одного или нескольких компонентов.

4. Количество необработанных сообщений в ключевых очередях брокера сообщений

Выполняется получение информации о конкретной очереди брокера сообщений:

- workflow.create.retry – очередь с сообщениями на повторную отправку потоков работ на создание;
- workflow.finish.retry – очередь с сообщениями на повторную отправку запросов на получение результатов обработки потоков работ;
- callback.send.retry – очередь с сообщениями на повторную отправку запросов с уведомлениями (webhook) об изменении состоянии потока работ.

Для получения информации, к Management API RabbitMQ отправляется HTTP-запрос:

```
GET /api/queues/{vhost}/{qname}
```

Где:

vhost – имя виртуального хоста, значение по умолчанию – docs;

qname – имя очереди.

Примечание. Для получения информации от Management API RabbitMQ требуется прохождение Basic авторизации.

Из полученного в ответ на запрос JSON-объекта берется значение `messages`, которое не должно превышать минимально возможное пороговое значение.

3.11 РЕКОМЕНДАЦИИ ПО МОНИТОРИНГУ МЕТРИК

3.11.1 ОБЩИЕ СВЕДЕНИЯ

В целях контроля работы сервиса, определения общего его состояния и выявления проблемных компонентов рекомендуется осуществлять мониторинг сервиса.

Для мониторинга разворачивается система мониторинга. В качестве средств мониторинга используются сторонние программные средства (например, Zabbix).

В данном руководстве не описан процесс установки и настройки программных средств мониторинга.

Примечание. Процесс установки, настройки и эксплуатации программных средств мониторинга, используемых в системе мониторинга, осуществляется в соответствии с документацией на них.

3.11.2 ОПТИМАЛЬНЫЙ НАБОР МЕТРИК ДЛЯ МОНИТОРИНГА

Мониторинг сервиса включает следующие процедуры и метрики:

1. Периодическая проверка (мониторинг) работы следующих компонентов TrustESS Docs (half-check):

- RabbitMQ;
- Redis;
- БД;
- файловое хранилище.

Основные метрики компонентов, требующие контроля указаны в **таблице 3.1**.

Таблица 3.1 – Основные метрики компонентов, требующие контроля

Наименование компонента	Метрика	Примечание
RabbitMQ (rabbit)	– контроль доступности компонента	Штатные средства RabbitMQ включают в себя REST API позволяющий следить за его доступностью и работоспособностью (half-check)
	– контроль увеличения числа запросов в очереди	Критическое число запросов устанавливается индивидуально в зависимости от объемов пользователя сервиса.
Redis	– контроль доступности компонента (half-check).	
БД (postgres)	– контроль доступности компонента, к примеру отправка тестового запроса <code>SELECT 1;</code>	
	– контроль наличия длительных запросов к компоненту.	Критическое число запросов устанавливается индивидуально в зависимости от объемов пользователя сервиса;
	– контроль блокировки запросов.	
Файловое хранилище	– контроль доступности компонента;	

	– проверка доступности компонента на чтение и запись файлов;	
	– контроль свободного места на компоненте.	

2. Периодическая проверка работы системы сервера, на котором развернут и функционирует TrustESS Docs

Осуществляется контроль:

- доступного места;
- загрузки ОЗУ;
- загрузки ЦПУ;
- и т.д.

Примечание. Перечень контролируемых параметров зависит от возможностей используемого программного обеспечения. Например, Zabbix имеет стандартные шаблоны для мониторинга.

3. Периодическая проверка доступности канала связи между TrustESS Docs и TrustESS Crypto.

4. Периодическая проверка доступности TLS GateWay.

❗ ВНИМАНИЕ!

1. В рамках использования компонента TLS GateWay в сценариях TrustESS Docs, рекомендуется проверять его доступность из открытой сети интернет.
2. Инструменты мониторинга компонента TLS GateWay предоставляются поставщиком оборудования.

3.12 РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ (НА)

Резервное копирование может осуществляться с использованием встроенных средств компонентов СД или сторонними средствами.

В СД подлежат обязательному резервному копированию следующие данные:

1. БД в СУБД PostgreSQL (<https://www.postgresql.org/docs/16/app-pg-dumpall.html>);
2. Сообщения в очередях брокера сообщений RabbitMQ (<https://www.rabbitmq.com/docs/backup#rabbitmq-messages>);
3. Файлы данных в файловом хранилище.

❗ ВНИМАНИЕ!

Для ускорения восстановления работоспособности СД после сбоев, рекомендуется иметь горячий резерв в виде реплик указанных данных на отдельных экземплярах компонентов.

4 ВОЗМОЖНЫЕ НЕПОЛАДКИ И СПОСОБЫ ИХ УСТРАНЕНИЯ

4.1 ОБЩИЕ СВЕДЕНИЯ

Для диагностики используются логи Docker-контейнеров, которые доступны напрямую на сервере через коллектор логов или команду:

```
docker logs <название контейнера>
```

Для изменения настроек требуется отредактировать файл с переменными окружения docs.env (по умолчанию расположен в директории: ./config/env) или конфигурационные файлы сервисов (Postgresql, RabbitMQ, Redis) подмонтированных снаружи.

После изменения настроек необходимо пересоздать Docker-контейнеры, для чего используется команда:

```
./manage.sh restart
```

или

```
docker-compose ... up -d --force-recreate.
```

4.2 ПОТЕРЯ СВЯЗНОСТИ КОНТЕЙНЕРОВ

4.2.1 ОПИСАНИЕ ОШИБКИ:

При обновлении или перезапуске части контейнеров Docker на ОС Astra Linux возможно возникновение проблемы с потерей соединения между контейнерами.

Проблема связана со сменой IP-адресов контейнеров и кешированием запросов к DNS на стороне ПО (например, в Nginx).

4.2.2 ВНЕШНЕЕ ПРОЯВЛЕНИЕ (СИМПТОМЫ)

1. При выполнении запросов к API сервиса возвращается ошибка HTTP 502 Bad Gateway, а все Docker-контейнеры находятся в состоянии Запущено (Up).
2. Web-интерфейс Панели администрирования доступен, но не получается авторизоваться. В консоли разработчика в браузере (F12 или Option+Command+I) на вкладке Сеть (Network) есть запросы с ошибкой HTTP 502 Bad Gateway.
3. Мониторинг может сообщать о недоступности сервиса.

4.2.3 РЕШЕНИЕ

Перезапуск Docker-контейнера с Nginx:

```
docker restart trust_entry-docs_frontend-1
```

```
# в некоторых версиях Docker наименование контейнера может отличаться:  
trust_entry_docs_frontend_1
```

4.3 НЕХВАТКА КОЛИЧЕСТВА СОЕДИНЕНИЙ С БД

4.3.1 ОПИСАНИЕ

При высокой нагрузке и/или долгой обработке запросов в БД новые API-запросы к сервису долго ожидают свободного соединения, либо получают ошибку превышения разрешенного количества соединений с БД.

4.3.2 ВНЕШНЕЕ ПРОЯВЛЕНИЕ (СИМПТОМЫ)

1. При выполнении запросов к API сервиса периодически или постоянно возвращается ошибка HTTP 500 Internal Error или HTTP 504 Gateway Timeout, при этом все Docker-контейнеры находятся в состоянии Запущено (Up).
2. В логах СУБД Postgresql наблюдаются сообщения:

```
FATAL: connection limit exceeded for non-superusers
```

4.3.3 РЕШЕНИЕ

1. Если ошибок в логах СУБД PostgreSQL нет, но при этом появляются таймауты при выполнении запросов к API в виде ошибок HTTP 504 Gateway Timeout, то необходимо увеличить параметр конфигурации сервиса DATABASE_POOL_SIZE до 10-30 (по умолчанию установлено значение 5) в файле docs.env.
2. Если ошибки в логах СУБД PostgreSQL есть, то необходимо увеличить количество соединений с БД (параметр конфигурации max_connections).
3. Для дальнейшего решения проблемы необходимо выгрузить из БД (pg_stat_statements или pg_stat_activity) длительные запросы для оптимизации.

ПАРАМЕТРЫ КОНФИГУРАЦИИ, ЗАДАВАЕМЫЕ В КОНФИГУРАЦИОННОМ ФАЙЛЕ DOCS.ENV

Параметр конфигурации	Описание	Значение по умолчанию
1. Параметры подключения к БД		
DATABASE_HOST	Адрес хоста с СУБД PostgreSQL	127.0.0.1
DATABASE_PORT	Порт хоста с СУБД PostgreSQL	5432
DATABASE_USER	Имя пользователя с БД для сервиса в СУБД PostgreSQL	docs
DATABASE_PASSWORD	Пароль пользователя с БД для сервиса в СУБД PostgreSQL	11111111
DATABASE_NAME	Имя БД для сервиса в СУБД PostgreSQL	docs
DATABASE_POOL_SIZE	Количество поддерживаемых открытыми соединений с БД	5
2. Параметры подключения к AMQP-брокеру (RabbitMQ)		
BROKER_URL	Адрес (URL) доступа к AMQP-брокеру (строка подключения)	amqp://guest:guest@rabbit:5672/docs
BROKER_API_PORT	Порт доступа к Management Console RabbitMQ для healthcheck	15672
3. Параметры подключения к Redis		
REDIS_URL	Адрес (URL) доступа к Redis	redis://10.0.36.121:6379/1

Параметр конфигурации	Описание	Значение по умолчанию
4. Параметры подключения к TrustESS CA и идентификаторы используемых сценариев		
ESS_CA_SERVER_URL	Адрес (URL) доступа к серверу TrustESS Ca с указанием протокола и порта	http://10.0.36.121:81
ESS_CA_SERVER_USERNAME	Имя пользователя для доступа к серверу TrustESS Ca	
ESS_CA_SERVER_PASSWORD	Пароль пользователя для доступа к серверу TrustESS Ca	
ESS_CA_SCENARIO_ISSUE_CERTIFICATE	Идентификатор сценария выпуска УНЭП на сервере TrustESS Ca	
ESS_CA_SCENARIO_ISSUE_CERTIFICATE_QUALIFIED	Идентификатор сценария выпуска УКЭП на сервере TrustESS Ca	
ESS_CA_SCENARIO_DOCUMENT_SIGN	Идентификатор сценария подписания УНЭП на сервере TrustESS Ca	
ESS_CA_SCENARIO_DOCUMENT_SIGN_QUALIFIED	Идентификатор сценария подписания УКЭП на сервере TrustESS Ca	
5. Служебные параметры		
ACCESS_TOKEN_EXPIRE_MINUTES	Продолжительность жизни токена авторизации (в минутах)	7 дней (60 * 24 * 7)
TIMEZONE	Временная зона, в которой учитываются даты	Europe/Moscow
LOG_LEVEL	Уровень логирования сервиса (Доступны стандартные уровни: ERROR, INFO, WARNING, DEBUG)	INFO
STATIC_DIR	Путь внутри контейнера до директории хранения файлов	static