

Инструкция по установке экземпляра ПО «Нормодок»

Содержание

1 Общие сведения	
2 Требования к среде развёртывания.....	3
3 Состав дистрибутива.....	5
4 Порядок установки.....	7
5 Первичная настройка.....	10
6 Проверка работоспособности.....	14
7 Контакты технического специалиста.....	17

1 Общие сведения

1.1 Настоящая инструкция устанавливает порядок установки экземпляра программного обеспечения «Нормодок» (далее - ПО) и предназначена для лиц, осуществляющих развёртывание ПО у правообладателя, у клиента либо в среде эксплуатации иным способом.

1.2 Правообладателем ПО является индивидуальный предприниматель Заварзин А. В.

1.3 ПО «Нормодок» представляет собой программный комплекс - ИИ-агент генерации и нормоконтроля технической документации по российским стандартам (ЕСПД, ЕСКД, ГОСТ 34/ГОСТ Р 59795, СРПП; реестр норм хранится и обрабатывается как структурированные данные, а не как текстовые шаблоны). ПО обеспечивает:

- генерацию документа требуемого типа по составу, заданному реестром норм;
- проведение нормоконтроля собственного или иного документа (проверка комплектности разделов, обозначения, оформления, структуры, актуальности ссылок на стандарты);

- формирование выходных файлов в форматах DOCX/PDF с оформлением по формам ГОСТ 2.104 и ГОСТ 19.104 (основная надпись, рамка, лист утверждения, титульный лист по ЕСПД).

1.4 ПО поставляется в виде набора программных компонентов (сервисов), объединённых в единый развёртываемый комплекс: приложение (веб-интерфейс), векторное хранилище Qdrant, реляционная база данных Supabase/PostgreSQL, объектное хранилище MinIO, графовая база данных Neo4j (граф норм). Взаимодействие пользователя с ПО осуществляется через веб-интерфейс.

1.5 Установка и запуск экземпляра ПО производятся средствами контейнеризации Docker и Docker Compose: все компоненты комплекса разворачиваются в виде набора контейнеров, управляемых единой конфигурацией docker-compose.

1.6 Экземпляр ПО устанавливается на сервере под управлением операционных систем семейства Linux, включая отечественные операционные системы Astra Linux и РЕД ОС; работа под управлением указанных ОС обеспечивается за счёт использования механизма контейнеризации Docker, изолирующего ПО от особенностей конкретного дистрибутива.

1.7 В состав ПО входит компонент искусственного интеллекта: большая языковая модель (LLM), используемая для генерации и анализа текста документации, и модуль локальных эмбеддингов, используемый для индексации и поиска по базе норм. Языковая модель может быть облачной либо развёрнутой локально - выбор варианта осуществляется клиентом при настройке экземпляра. Модуль эмбеддингов работает локально, без передачи обрабатываемых данных вовне.

1.8 Техническую поддержку экземпляра ПО осуществляет правообладатель - индивидуальный предприниматель Заварзин А. В. Контактные данные для обращения по вопросам технической поддержки указаны на официальном сайте

нормодок.рф; адрес электронной почты службы технической поддержки - support@нормодок.рф.

1.9 Настоящая инструкция описывает порядок установки экземпляра ПО и не заменяет иные документы эксплуатационной документации (руководство администратора, руководство пользователя), а также не содержит сведений о стоимости, порядке приобретения и условиях предоставления ПО.

2 Требования к среде развёртывания

2.1 Экземпляр ПО «Нормодок» развёртывается на сервере под управлением операционной системы семейства Linux, в том числе отечественных дистрибутивов Astra Linux и РЕД ОС; совместимость с указанными ОС обеспечивается за счёт контейнеризации всех компонентов средствами Docker (запуск через Docker Compose без прямой зависимости от специфики дистрибутива).

2.2 На сервере должны быть установлены Docker Engine и Docker Compose версий, поддерживающих используемый в составе дистрибутива формат compose-файла: Docker Engine версии 25 или новее и Docker Compose v2 (совместимость подтверждена протоколами испытаний на РЕД ОС 8 (Docker Engine 29.x) и Astra Linux 1.8 (Docker Engine 25.0)).

2.3 Развёртывание выполняется единым стеком контейнеров docker-compose, включающим следующие сервисы:

- app - основное приложение (веб-интерфейс на базе Chainlit), реализующее функции генерации документов и нормоконтроля;
- Qdrant - векторное хранилище для локальных эмбедингов и поиска по реестру норм и эталонам;
- Supabase (PostgreSQL) - реляционное хранилище данных приложения;
- MinIO - объектное хранилище файлов (исходные документы, результаты рендеринга DOCX/PDF, эталоны форм);

- Neo4j - графовая база данных для хранения графа норм и связей между требованиями.

2.4 Доступ пользователей к экземпляру ПО осуществляется через веб-интерфейс (браузер) по протоколу HTTP(S); установка отдельного клиентского программного обеспечения на рабочих местах пользователей не требуется.

2.5 Сетевые порты сервисов публикуются только на локальном интерфейсе сервера (localhost); конкретные номера портов веб-интерфейса и служебных сервисов определяются конфигурацией развёртывания. Типовая карта портов: веб-интерфейс - 8000/tcp; Qdrant - 6333/tcp; PostgreSQL - 5432/tcp; MinIO - 9100/tcp и 9101/tcp; Ollama (локальная LLM) - 11434/tcp. Внешний доступ предоставляется только к веб-интерфейсу через обратный прокси-сервер.

2.6 Для функционирования подсистемы искусственного интеллекта требуется доступ к большой языковой модели (LLM): по выбору заказчика используется облачная LLM (через внешний шлюз/API) либо локальная LLM, развёрнутая в инфраструктуре заказчика. Локальные эмбединги для индексации и поиска рассчитываются средствами приложения без передачи данных во внешние сервисы.

2.7 При использовании облачной LLM требуется исходящий доступ сервера к сети Интернет и наличие действующих реквизитов доступа (API-ключей) к выбранному провайдеру. При использовании локальной LLM исходящий доступ в Интернет для работы подсистемы генерации и нормоконтроля не требуется.

2.8 Аппаратные ресурсы сервера (число процессорных ядер, объём оперативной памяти, объём дискового пространства, наличие графического ускорителя для расчёта локальных эмбедингов) определяются исходя из нагрузки и выбранной конфигурации ИИ-подсистемы (облачная/локальная LLM). Минимальная конфигурация: 4 процессорных ядра, 8 ГБ оперативной памяти, 40 ГБ дискового пространства; рекомендуемая: 8 ядер, 16 ГБ, 100 ГБ на SSD-накопителе.

Для автономного режима с локальной LLM - не менее 16 ГБ оперативной памяти; графический ускоритель необязателен (при использовании - от 8 ГБ видеопамяти).

2.9 Для хранения данных сервисов (Qdrant, Supabase, MinIO, Neo4j) и результатов работы приложения на сервере должно быть выделено персистентное дисковое пространство, подключаемое к контейнерам в виде именованных томов и монтируемых каталогов.

2.10 Устанавливаемое программное обеспечение не требует наличия проприетарного системного ПО, отсутствующего в свободном доступе; для формирования документов по формам ГОСТ 2.104 и ГОСТ 19.104 в контейнере приложения предустановлены свободно распространяемые шрифтовые метрические аналоги, обеспечивающие корректное отображение и печать основной надписи и рамки.

2.11 Техническая поддержка экземпляра ПО в части требований к среде развёртывания и содействия при установке осуществляется правообладателем - ИП Заварзин А.В.; адрес электронной почты - support@нормодок.рф, форма обратной связи - на сайте нормодок.рф.

3 Состав дистрибутива

3.1 Дистрибутив экземпляра ПО «Нормодок» поставляется в виде набора файлов конфигурации и исходного кода для развёртывания средствами Docker Compose и включает следующие компоненты.

3.2 Файл конфигурации развёртывания docker-compose.yml, описывающий состав и параметры запуска сервисов единого стека на базе образа normodoc-app: приложение пользователя app (веб-интерфейс Chainlit, порт 8765), кабинет владельца admin (порт 8799), кабинет нормоконтролёров reviewer (порт 8788), сервис приёма заявок leads (порт 8790), шлюз локальной LLM llm-gateway (порт 8794), а также сопутствующие сервисы: qdrant (векторное хранилище, порт 6343),

supabase/PostgreSQL (статусный фонд, очереди, порт 54329), minio (хранилище PDF-копий норм, порты 9100/9101), neo4j (граф замен стандартов, порты 7475/7688).

3.3 Файл-образец переменных окружения `.env.example` для формирования рабочего файла `.env` (ключи доступа к LLM-провайдеру, параметры подключения к сервисам, пороги нормоконтроля и прочие настраиваемые параметры).

3.4 Исходный код приложения на языке Python 3.10+, организованный по подсистемам: `src/agent/` (состояние агента, граф обработки, узлы, промпты), `src/norms/` (реестр норм, детерминированные проверки, статусный фонд, самопополнение корпуса), `src/kit/` (сборка и проверка комплектов документов), `src/drawings/` (нормоконтроль текстовой части чертежей), `src/revision/` (обработка правок по замечаниям), `src/reviewers/` (кабинет выверки реестра), `src/recon/` (сверка реестра с официальными источниками), `src/render/` (формирование DOCX/PDF), `src/parsers/` (чтение входных документов), `src/llm/`, `src/embeddings/`, `src/vectorstore/` (провайдеры моделей и хранилищ), `src/admin/` (модули административного кабинета), `src/storage/` (хранение истории обращений).

3.5 Библиотека выверенных эталонов форм оформления документов `src/render/frames/`, обеспечивающая рендеринг DOCX/PDF с рамкой, основной надписью, листом утверждения и титульным листом по формам ГОСТ 2.104 и ГОСТ 19.104.

3.6 Каталог данных реестра норм `src/norms/data/*.yaml` с описанием состава разделов, обозначений, редакций по датам и профилей документов по стандартам ЕСПД, ГОСТ 34, ЕСКД, СРПП.

3.7 Каталог эталонных примеров `examples/` для формирования библиотеки образцов, используемой при генерации документов.

3.8 Файл описания зависимостей Python `requirements.txt`, необходимый для установки в хост-режиме (режим разработки) без использования Docker.

3.9 Файл управления сборкой и командами эксплуатации Makefile с целевыми командами: make up (запуск стека), make index (индексация эталонов и текстов норм в Qdrant), make run (запуск веб-интерфейса), make test (автоматические тесты), make lint (статическая проверка кода), make deploy (пересборка образа и перезапуск стеков), make up-gpu (оверлей для использования GPU при вычислении локальных эмбедингов).

3.10 Скрипт создания клиентских (тенантских) стеков scripts/tenant.sh для мультитенантного развёртывания экземпляров ПО.

3.11 Сопроводительная документация: служебная документация разработчика в составе дистрибутива (порядок установки и эксплуатации, описание архитектуры, порядок добавления типа документа или стандарта в реестр норм, описание работы с комплектами документов).

3.12 Средства локальной обработки данных: конфигурация локальных эмбедингов (FastEmbed, multilingual-e5-large) и, при выборе клиентом локального варианта ИИ, конфигурация подключения к локальной LLM (Ollama/vLLM) через шлюз llm-gateway; при выборе облачного варианта - конфигурация подключения к облачному LLM-провайдеру через LangChain/OpenRouter.

3.13 Перечень файлов дистрибутива и их контрольные суммы (MD5/SHA-256) передаются вместе с дистрибутивом. Дистрибутив передаётся конечному пользователю защищённой ссылкой на архив после заключения договора; по согласованию сторон возможна передача на физическом носителе.

4 Порядок установки

4.1. Установка экземпляра программного обеспечения «Нормодок» выполняется на сервере под управлением операционной системы Linux, в том числе российских ОС Astra Linux и РЕД ОС, при помощи технологии контейнеризации Docker и инструмента Docker Compose. Все компоненты ПО (веб-приложение,

векторное хранилище Qdrant, СУБД Supabase/PostgreSQL, объектное хранилище MinIO, граф-СУБД Neo4j) запускаются из единого docker-compose стека одной командой, без отдельной установки каждого компонента.

4.2. Перед началом установки необходимо убедиться в наличии на сервере предустановленных Docker и Docker Compose, а также в выполнении требований к аппаратным и программным средствам, приведённых в разделе «Требования к серверу и рабочим местам пользователей» (Docker Engine 25 или новее, Docker Compose v2).

4.3. Для установки необходимо получить дистрибутив ПО «Нормодок» (архив с исходным кодом и конфигурационными файлами Docker Compose), переданный правообладателем защищённой ссылкой после заключения договора, и разместить его в рабочем каталоге на сервере.

4.4. Установка выполняется в следующем порядке:

4.4.1. Скопировать файл примера настроек окружения в рабочий файл настроек командой: `cp .env.example .env`

4.4.2. Заполнить в полученном файле настроек параметры подключения к используемым сервисам, а также параметры выбора провайдера ИИ (LLM_PROVIDER): облачная LLM (через OpenRouter, с указанием ключа доступа) либо локальная LLM (через Ollama/vLLM на стороне заказчика) - выбор осуществляется клиентом самостоятельно. Локальные эмбединги устанавливаются и работают в составе поставляемого образа без обращения к внешним сервисам.

4.4.3. Выполнить развёртывание и запуск всех сервисов стека (векторное хранилище Qdrant, СУБД Supabase/PostgreSQL, объектное хранилище MinIO, граф-СУБД Neo4j, веб-приложение) командой: `make up` что соответствует выполнению команды `docker compose up -d --build` со сборкой образа приложения.

4.4.4. При необходимости использования графического процессора для локальных эмбеддингов выполнить дополнительный запуск с GPU-оверлеем командой `make up-gpu`, предварительно установив `nvidia-container-toolkit` и выполнив настройку среды выполнения Docker для работы с GPU. Без установки данного компонента эмбеддинги выполняются на центральном процессоре сервера.

4.4.5. Выполнить индексацию эталонных документов и текстов нормативных документов в векторном хранилище Qdrant командой: ``` make index`

4.4.6. Проверить состояние запущенных контейнеров командой `make ps`; при необходимости просмотра диагностических сведений использовать команду `make logs`.

4.4.7. Убедиться в доступности веб-интерфейса пользователя (Chainlit) по адресу `http://127.0.0.1:8765` (порт указан для локального обращения; при необходимости внешнего доступа настраивается проксирование средствами заказчика).

4.4.8. При необходимости проверки работоспособности программных компонентов выполнить команду автоматического тестирования `make test`, а также команду статического контроля качества кода `make lint`.

4.5. Для обновления версии установленного экземпляра ПО выполняется команда `make deploy`, обеспечивающая пересборку образа приложения и перезапуск сервисов стека.

4.6. По результатам установки должны быть доступны: веб-интерфейс пользователя для выполнения режимов работы ПО, кабинет нормоконтролёра, кабинет администратора и служебные сервисы (векторное хранилище, СУБД, объектное и графовое хранилища), взаимодействующие между собой в рамках единого сетевого пространства `docker-compose`.

4.7. Установка считается завершённой при успешном запуске всех контейнеров стека, отсутствии ошибок в диагностических сведениях и доступности веб-интерфейса пользователя по указанному адресу.

4.8. Техническая поддержка по вопросам установки экземпляра ПО осуществляется правообладателем - ИП Заварзиным А.В. - по контактными данным, указанным на официальном интернет-сайте нормодок.рф; адрес электронной почты - support@нормодок.рф.

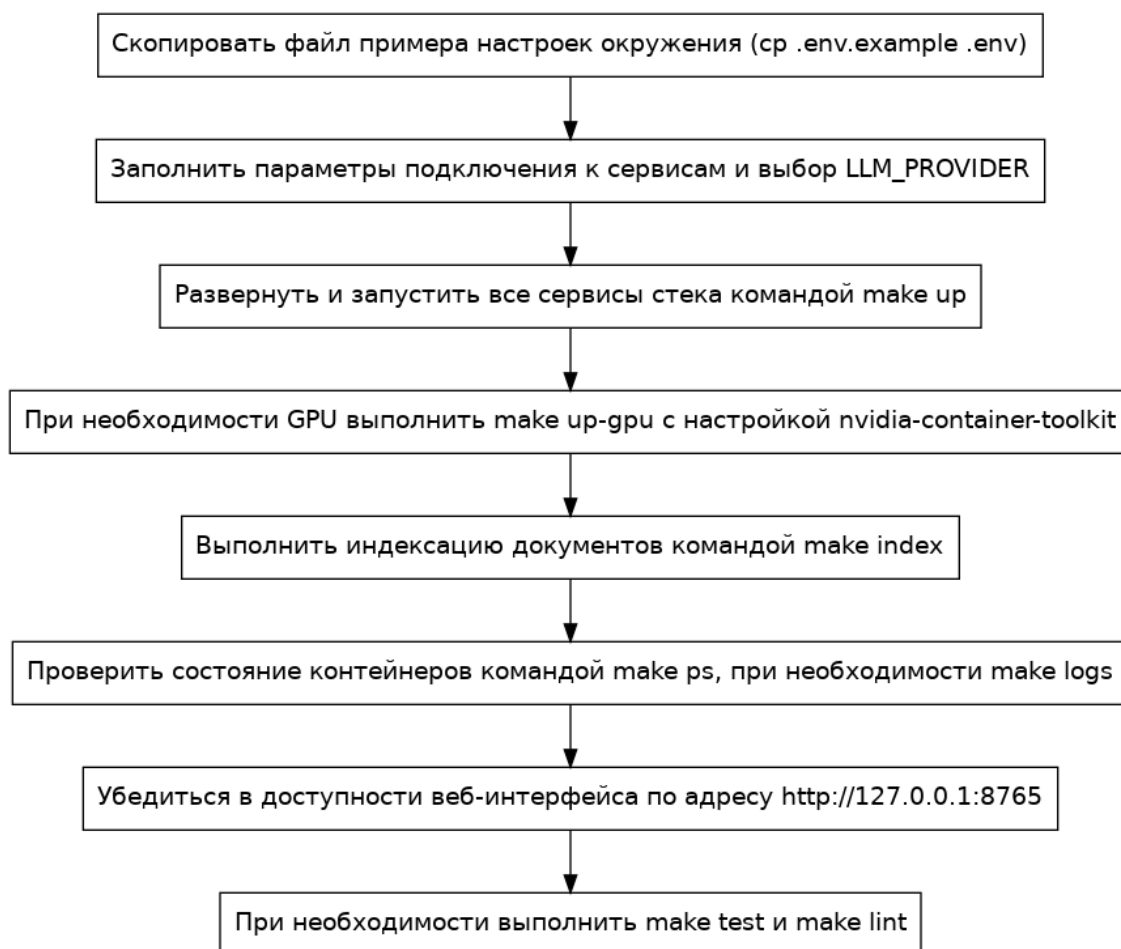


Рисунок 1 - Порядок действий: Порядок установки (схема построена по тексту документа)

5 Первичная настройка

5.1 После установки экземпляра ПО «Нормодок» (докер-образ развёрнут, контейнеры app, Qdrant, Supabase/PostgreSQL, MinIO и Neo4j запущены и находятся

в состоянии healthy) выполняется первичная настройка экземпляра перед началом эксплуатации.

5.2 Настройка конфигурационного файла окружения. На основе поставляемого шаблона `.env.example` создаётся рабочий файл `.env`, в котором заполняются следующие параметры:

- реквизиты доступа к используемой языковой модели (ИИ): при выборе облачного варианта - ключ доступа к сервису-провайдеру LLM (переменная `OPENROUTER_API_KEY` или аналогичная в зависимости от выбранного шлюза); при выборе локального варианта - адрес локального сервиса инференса (Ollama/vLLM) и наименование модели;
- наименование используемой модели ИИ (`OPENROUTER_MODEL` или соответствующая переменная локального шлюза);
- параметры модели локальных эмбеддингов (`EMBEDDINGS_MODEL`);
- адрес подключения к сервису Qdrant (`QDRANT_URL`);
- параметры подключения к Supabase/PostgreSQL, MinIO и Neo4j (адреса, порты, учётные данные), формируемые автоматически при первом запуске либо задаваемые администратором вручную в конфигурационном файле `.env` (в типовой установке ручное задание не требуется);
- пороговое значение прохождения нормоконтроля (`NORMCONTROL_PASS_THRESHOLD`);
- максимальное число циклов доработки документа в графе «генерация - нормоконтроль» (`MAX_REVISIONS`);
- используемый движок формирования PDF (`PDF_ENGINE`: WeasyPrint либо LibreOffice headless);
- каталог вывода результатов рендеринга (`RENDER_OUTPUT_DIR`).

Файл `.env` не подлежит передаче третьим лицам и не включается в системы контроля версий.

5.3 Выбор варианта использования ИИ. Администратор экземпляра определяет режим работы модуля ИИ:

- облачная LLM - с указанием провайдера и ключа доступа в файле .env;
- локальная LLM - с указанием адреса локального шлюза инференса (Ollama/vLLM) и модели, развёрнутой на стороне заказчика.

Выбор варианта не влияет на состав и работу локальных эмбедингов (FastEmbed, multilingual-e5-large), которые исполняются локально независимо от выбранного варианта ИИ и не передают данные за пределы контура заказчика.

5.4 Первичная индексация данных. После заполнения .env и запуска сервиса Qdrant выполняется команда индексации эталонов форм и текстов норм в векторное хранилище (make index). Индексация выполняется однократно при первичной настройке и повторно - при обновлении библиотеки эталонов (src/render/frames/) или реестра норм.

5.5 Проверка готовности сервисов. Состояние контейнеров стека проверяется командой просмотра статусов (make ps); при необходимости диагностики используется просмотр журналов (make logs). Веб-интерфейс становится доступен по адресу, заданному для сервиса Chainlit (порт 8765, доступ ограничен локальным контуром - 127.0.0.1, либо контуром сети заказчика согласно его настройкам сетевой безопасности).

5.6 Настройка шрифтов оформления. В образе используются свободные метрические аналоги шрифтов Times New Roman/Arial (Liberation Serif/Sans). Для обеспечения пиксель-точного соответствия форм ГОСТ 2.104 и ГОСТ 19.104 эталонным шаблонам администратор экземпляра может дополнительно установить в образ шрифты Times New Roman/Arial - данное действие является необязательным.

5.7 Настройка графического модуля. Для формирования блок-схем в составе документации требуется наличие в контуре развёртывания пакета Graphviz (утилита dot).

5.8 Настройка чтения входных документов. При первом обращении к режимам, требующим распознавания входных PDF-документов и сканов (модуль docling с локальным OCR rapidocr), выполняется автоматическая загрузка моделей распознавания (детектор общего назначения, модуль распознавания текста для славянских языков). Первый запрос по данному режиму выполняется со значительной задержкой, обусловленной загрузкой моделей; последующие обращения используют локальный кэш. Распознавание сканов выполняется локально, без передачи данных за пределы контура заказчика.

5.9 Настройка мультитенантности (при развёртывании для нескольких клиентских контуров). Создание отдельного клиентского стека выполняется командой `scripts/tenant.sh create <имя> [--cloud --model <модель>]`, где параметр `--cloud` определяет использование облачной модели ИИ, а параметр `--model` - конкретную модель. Порядок обновления версий клиентских стеков выполняется командой `make deploy` (пересборка образа, перезапуск стеков, обновление тенантов).

5.10 Настройка графического интерфейса пользователя (Chainlit) не требует отдельных действий после выполнения пунктов 6.2–6.5: интерфейс запускается автоматически в составе контейнера `app` и предоставляет доступ ко всем режимам работы («Один документ», «Нормоконтроль», «Нормоконтроль чертежа», «Комплект документов», «Корпоративный профиль», «Титульный лист», «Импорт формы», «Доработка по замечаниям», «Задать вопрос»).

5.11 После выполнения пунктов 6.1–6.10 экземпляр ПО «Нормодок» считается готовым к проверке работоспособности в соответствии с разделом «Проверка работоспособности» настоящей инструкции. Обращения по вопросам первичной

настройки направляются в службу технической поддержки правообладателя (ИП Заварзин А. В.); адрес электронной почты - support@нормодок.рф.

6 Проверка работоспособности

6.1 После выполнения команд установки, приведённых в разделах настоящей инструкции (make up - развёртывание контейнеров Docker Compose, make index - индексация эталонов и текстов норм в Qdrant), необходимо выполнить проверку работоспособности экземпляра ПО «Нормодок» в следующем порядке.

6.2 Проверить состояние контейнеров стека командой make ps (или docker compose ps). Все сервисы - app (Chainlit), qdrant, supabase_db, minio, neo4j - должны находиться в состоянии «running» (up), без перезапусков и ошибок завершения. Признак неисправности - статус «restarting» или «exited» у любого из сервисов; в этом случае следует обратиться к журналам (make logs) для диагностики.

6.3 Проверить доступность веб-интерфейса пользователя (Chainlit): в браузере на сервере или в локальной сети открыть адрес <http://127.0.0.1:8765> (либо адрес, назначенный при развёртывании). Признак работоспособности - загрузка стартового экрана с карточками режимов работы: «Один документ», «Нормоконтроль», «Нормоконтроль чертежа», «Комплект документов», «Корпоративный профиль», «Титульный лист», «Импорт формы», «Доработка по замечаниям», «Задать вопрос».

6.4 При наличии в составе развёртывания дополнительных сервисов проверить их доступность аналогичным образом: кабинет нормоконтролёра - порт 8788, административный интерфейс - порт 8799. Доступ к перечисленным портам предоставляется только на localhost (или в пределах локальной сети заказчика по согласованной схеме доступа) в соответствии с принятой политикой безопасности.

6.5 Проверить подключение к хранилищу векторных представлений: выполнить запрос состояния сервиса Qdrant (порт 6343) средствами администратора СУБД или встроенным инструментом диагностики; признак работоспособности -

успешный ответ сервиса и наличие проиндексированных коллекций (эталонов форм и текстов норм), созданных командой `make index`.

6.6 Проверить подключение к реляционному хранилищу Supabase/PostgreSQL (порт 54329) и к объектному хранилищу MinIO (порты 9100/9101): сервисы должны принимать соединения и не завершаться с ошибкой при обращении из контейнера `app`.

6.7 Проверить подключение к графовой базе данных Neo4j (порты 7475/7688), используемой для хранения реестра норм в виде графа связей между нормативными документами и разделами.

6.8 Выполнить функциональную проверку через веб-интерфейс: в режиме «Один документ» задать тип документа и минимальный источник (описание объекта автоматизации) и убедиться, что агент формирует документ с заданным составом разделов, проводит нормоконтроль и предоставляет отчёт с провенансом (ссылками на использованные пункты нормы). Работоспособность подтверждается получением на выходе файла в формате DOCX или PDF с оформлением по формам ГОСТ 2.104 / ГОСТ 19.104 (рамка, основная надпись).

6.9 Выполнить проверку режима «Нормоконтроль»: загрузить готовый документ (.docx, .pdf, .txt или .md) и убедиться, что агент выполняет комплектность разделов, проверку обозначения, оформления и структуры документа, формируя отчёт о результатах.

6.10 Проверить корректность работы модуля распознавания входных документов: при загрузке PDF-документа без текстового слоя (скана) убедиться, что распознавание выполняется локальным средством оптического распознавания (RapidOCR) без передачи данных во внешние сервисы. При первом обращении к такому документу допускается увеличенное время отклика, связанное с загрузкой моделей распознавания в локальный кэш; последующие обращения выполняются из кэша.

6.11 Проверить взаимодействие с компонентом искусственного интеллекта (языковой моделью): в зависимости от выбранной клиентом конфигурации - облачной (через шлюз OpenRouter) или локальной (Ollama/vLLM) - убедиться в успешном получении ответа при выполнении генерации или нормоконтроля документа. Признак неисправности - ошибка соединения с моделью или отсутствие ответа в течение времени, установленного конфигурацией.

6.12 При использовании графического процессора (опциональный оверлей `make up-gpu`) проверить, что вычисление локальных эмбедингов (FastEmbed, multilingual-e5-large) выполняется на CUDA; при отсутствии совместимого драйвера или GPU эмбединги автоматически выполняются на центральном процессоре - это не является признаком неисправности.

6.13 Выполнить автоматизированную проверку внутренней логики компонентов ПО командой `make test` (набор автоматических тестов). Успешное завершение проверки (без ошибок) подтверждает корректность работы основных программных модулей экземпляра ПО.

6.14 Экземпляр ПО «Нормодок» считается работоспособным при одновременном выполнении условий, указанных в пунктах 6.2–6.13: все контейнеры стека запущены и устойчивы, веб-интерфейс и связанные с ним компоненты доступны, подключения к хранилищам данных установлены, функции генерации и нормоконтроля документов выполняются с получением результата в установленном формате, автоматизированные тесты пройдены без ошибок.

6.15 При выявлении отклонений от указанных признаков работоспособности следует обратиться к журналам сервисов (`make logs`) и к документу по эксплуатации системы, а при невозможности устранения неисправности своими силами - в службу технической поддержки правообладателя (ИП Заварзин А. В.); адрес электронной почты - support@нормодок.рф.

7 Контакты технического специалиста

7.1 Техническую поддержку экземпляра программного обеспечения «Нормодок» осуществляет правообладатель - индивидуальный предприниматель Заварзин А. В.

7.2 Обращения по вопросам установки, настройки и эксплуатации экземпляра ПО принимаются по контактными данным, указанным на официальном сайте нормодок.рф: адрес электронной почты - support@нормодок.рф; также доступна форма обратной связи на сайте.

7.3 Официальные реквизиты правообладателя: ИНН 667409480434, ОГРНИП 307667415100046; актуальные сведения размещаются на сайте нормодок.рф.

7.4 Режим работы технической поддержки: рабочие дни, с 10:00 до 19:00 по московскому времени; ответ на обращение - не позднее следующего рабочего дня.

7.5 При обращении технический специалист принимает от пользователя следующие сведения, необходимые для диагностики: версию (номер сборки) экземпляра ПО «Нормодок», используемую операционную систему сервера (Linux, в том числе Astra Linux, РЕД ОС), способ развёртывания (Docker Compose), состав задействованных сервисов (Qdrant, Supabase/PostgreSQL, MinIO, Neo4j, приложение app), а также журналы (логи), полученные командой make logs, и текст возникшей ошибки.

7.6 Информирование пользователей об обновлениях, известных ошибках и рекомендациях по установке осуществляется на странице документации официального сайта (нормодок.рф/docs/) и по электронной почте по запросу пользователя.

7.7 Иные каналы связи: форма обратной связи на официальном сайте нормодок.рф; обращения регистрируются в журнале обращений правообладателя.