SANDBOXER

ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ

Инструкция по установке экземпляра программного обеспечения «TaskFlow AI»

Внутреннее (рабочее) наименование, используемое в исходном коде, — Sandboxer Mail Router

Правообладатель	ООО «СЭНДБОКСЕР»
ОГРН / ИНН	1247700378562 / 9718258965
Версия документа	1.0
Дата	24 сентября 2026 г.

107113, г. Москва, ул. 3-я Рыбинская, д. 18, стр. 2 · sandboxer.ru · contact@sandboxer.ru

1. Назначение документа

Документ описывает порядок установки экземпляра программного обеспечения «TaskFlow AI» (далее — ПО) в информационной инфраструктуре заказчика, требования к аппаратному и программному обеспечению, порядок первичной настройки, проверки работоспособности, обновления и восстановления.

Документ предназначен для системных администраторов и инженеров эксплуатации заказчика, а также для проведения экспертной проверки экземпляра ПО.

2. Условия и способ поставки

ПО поставляется в виде экземпляра, устанавливаемого в инфраструктуре заказчика (on-premise) либо на выделенном сервере, предоставляемом правообладателем по договору. Модель распространения в виде публичного интернет-сервиса с самостоятельной регистрацией пользователей не применяется; специальные действия по установке на рабочих местах конечных пользователей не требуются — работа выполняется в браузере.

Дистрибутив представляет собой набор контейнерных образов и файлов конфигурации. Публичное скачивание дистрибутива не предусмотрено: доступ к реестру контейнерных образов правообладателя предоставляется заказчику после заключения договора — выдаются учётные данные (логин и токен доступа) с ограниченными правами на чтение. По требованию заказчика, эксплуатирующего изолированный контур, дистрибутив передаётся архивом контейнерных образов на носителе или по защищённому каналу связи.

Развёртывание и первичную настройку экземпляра выполняют инженеры правообладателя либо инженеры заказчика по настоящей инструкции.

3. Состав программного обеспечения

Компонент	Назначение	Состав
Служба программного интерфейса mail-router-api	Серверная часть	REST API, аутентификация OIDC, управление ящиками, правилами, коннекторами и пользователями, канал WebSocket. Язык Python, фреймворк FastAPI
Служба обработки mail-router-worker	Серверная часть	Конвейер обработки сообщений: получение почты, классификация, извлечение данных, маршрутизация, исполнение действий
Планировщик mail-router-scheduler	Серверная часть	Периодические задачи: опрос почтовых ящиков, контроль SLA, напоминания, очистка
Брокер сообщений NATS JetStream	Инфраструктура	Очередь обработки сообщений с гарантией доставки и сохранением на диск
Векторное хранилище Qdrant	Инфраструктура	Векторные представления сообщений для сопоставления с историческим контекстом
Веб-клиент mail-router-ui	Клиентская часть	Пользовательский интерфейс, статическая сборка, отдаётся веб-сервером nginx

Реляционная СУБД PostgreSQL и S3-совместимое хранилище вложений (MinIO) могут использовать существующие инстансы заказчика либо разворачиваться в составе установки. Аутентификация выполняется через провайдера OIDC заказчика (в том числе Keycloak).

4. Требования к аппаратному обеспечению

Параметр	Минимально	Рекомендуется
Процессор	4 ядра x86-64	8 ядер x86-64 и выше; при размещении локальной языковой модели в контуре — от 16 ядер
Оперативная память	8 ГБ	16 ГБ и выше; при локальной языковой модели — от 64 ГБ
Дисковое пространство	50 ГБ SSD	100 ГБ и более SSD (зависит от объёма почты и вложений)
Графический ускоритель	Не требуется: модели исполняются на стороне шлюза языковых моделей либо на процессоре в контуре заказчика	

Указанные требования относятся к серверу приложений. При размещении PostgreSQL, NATS, Qdrant и S3-хранилища на том же сервере объём дискового пространства увеличивается пропорционально объёму хранимой почты и вложений (ориентировочно 1 ГБ на 10 000 обработанных сообщений с вложениями).

5. Требования к программному обеспечению

Компонент	Требование
Операционная система	Linux x86-64 с ядром 5.10 и выше: Ubuntu Server 22.04/24.04 LTS, Debian 12, Astra Linux Special Edition 1.7, «Ред ОС» 7.3 и другие дистрибутивы с поддержкой контейнеризации
Среда контейнеризации	Docker Engine 24.0 и выше, Docker Compose v2
СУБД	PostgreSQL 16 и выше (рекомендуется 18)
Обратный прокси	Traefik 3 или nginx с поддержкой TLS и WebSocket
Провайдер аутентификации	OIDC-совместимый провайдер (Keycloak 24 и выше либо корпоративный провайдер заказчика)
Языковая модель	Доступ к шлюзу больших языковых моделей, совместимому с интерфейсом OpenAI. Допускается развёртывание модели в контуре заказчика
Почтовая инфраструктура	Доступ к почтовым серверам: EWS (443), IMAPS (993), POP3S (995), SMTP (587)

5.1. Сетевые требования

- Входящие соединения: TCP 443 (HTTPS, включая WebSocket) — веб-интерфейс и программный интерфейс; TCP 80 — перенаправление на HTTPS и выпуск сертификатов.
- Внутренние соединения: TCP 8014 — служба программного интерфейса, TCP 5432 — PostgreSQL, TCP 14222 — NATS, TCP 6333/6334 — Qdrant, TCP 9000 — S3-хранилище. Перечисленные порты не публикуются наружу: службы доступны только через обратный прокси и внутри контура установки.
- Исходящие соединения на этапе установки: доступ к реестру контейнерных образов правообладателя. Для изолированного контура — не требуется, образы передаются архивом.
- Исходящие соединения при эксплуатации: доступ к почтовым серверам организации, к шлюзу языковых моделей, к подключаемым внешним системам (Битрикс24, Directum, мессенджеры и другие коннекторы по требованиям этих систем).

6. Порядок установки

Шаг 1. Подготовка сервера

Установить операционную систему и среду контейнеризации, создать служебного пользователя и рабочий каталог:

```
sudo mkdir -p /opt/sandboxer/mail-router
cd /opt/sandboxer/mail-router
```

Шаг 2. Получение дистрибутива

Авторизоваться в реестре контейнерных образов правообладателя с использованием выданных учётных данных и загрузить образы:

```
docker login registry.sandboxer.ru
docker compose -f docker-compose.prod.yml pull
```

Для изолированного контура образы загружаются из переданного архива:

```
docker load -i sandboxer-mail-router-<версия>.tar
```

Шаг 3. Подготовка внешних сервисов

Создать базу данных и учётную запись СУБД. Если внешние сервисы разворачиваются в контейнерах, используются описания из состава дистрибутива.

```
CREATE DATABASE mail_router;
```

Шаг 4. Настройка переменных окружения

Скопировать файл-образец env.example в .env и заполнить значения. Обязательные группы параметров:

Группа	Параметры
Служба	Имя службы, сетевой адрес и порт, публичное доменное имя, уровень журналирования, часовой пояс
База данных	Строка подключения к PostgreSQL, схема
Брокер сообщений	Адрес и порт NATS, имя потока (stream)
Векторное хранилище	Адрес и порт Qdrant, ключ доступа, наименование коллекции
Хранилище вложений	Параметры доступа к S3-совместимому хранилищу, имя корзины
Аутентификация	Адрес OIDC-провайдера, идентификатор и секрет клиента, ключ шифрования сессионных cookie, ключ шифрования учётных данных ящиков
Языковая модель	Адрес шлюза, ключ доступа, наименование основной и резервной модели, лимиты запросов и токенов, тайм-аут
Почтовые ящики	Протоколы по умолчанию, интервалы опроса, лимиты выборки
Коннекторы	Адреса и ключи доступа подключаемых внешних систем (при использовании)

Внимание. Файл .env содержит секреты (ключи шифрования, пароли СУБД, ключи доступа). Права доступа к файлу должны быть ограничены служебным пользователем: chmod 600 .env .

Шаг 5. Запуск контейнеров

```
docker compose -f docker-compose.prod.yml up -d
```

Шаг 6. Применение миграций базы данных

```
docker compose -f docker-compose.prod.yml exec fastapi-app alembic upgrade head
```

Шаг 7. Настройка обратного прокси и TLS

Настроить публикацию веб-интерфейса и программного интерфейса на доменном имени заказчика, установить сертификат TLS. Правила маршрутизации (включая передачу WebSocket) входят в состав дистрибутива.

Шаг 8. Проверка работоспособности

Выполнить проверки раздела 8.

7. Первичная настройка

1. Зарегистрировать клиента ПО в OIDC-провайдере заказчика, указать адреса перенаправления, получить идентификатор и секрет клиента; заполнить параметры аутентификации в .env и перезапустить службу.
2. Создать роли пользователей и назначить их сотрудникам: администратор, оператор, пользователь, наблюдатель.
3. Подключить первый почтовый ящик: в разделе управления ящиками указать адрес, протокол (EWS, IMAP или POP3), учётные данные и параметры опроса. Убедиться, что входящие сообщения появляются в списке и проходят конвейер обработки.
4. Настроить категории обращений и глобальные правила маршрутизации под процессы заказчика; при необходимости — матрицу маршрутизации и SLA-профили.
5. Подключить коннекторы целевых систем (Битрикс24, Directum RX, 1С и другие), настроить сопоставление полей.
6. Подготовить шаблоны ответов на русском и английском языках.

8. Проверка работоспособности установки

1. Проверка состояния службы — ожидается ответ со статусом 200 OK:

   ```
   curl -f https://<домен>/health
   ```

2. Проверка готовности (зависимости: база данных, брокер, векторное хранилище):

   ```
   curl -f https://<домен>/ready
   ```

3. Тестовое сообщение: направить письмо на подключённый почтовый ящик. Ожидаемый результат: письмо появляется в списке входящих через интервал опроса, ему присваиваются категория и приоритет, извлекаются структурированные данные,

выполняется назначенный правилом маршрут; запись о каждом шаге видна в журнале обработки.

4. Проверка веб-интерфейса: открыть публичный адрес в браузере, выполнить вход через корпоративную учётную запись, убедиться в отображении дашборда и списка сообщений.
5. Проверка канала событий: оставить страницу дашборда открытой и направить тестовое письмо — счётчики должны обновиться без перезагрузки страницы.

9. Обновление версии и откат

1. Создать резервную копию базы данных и файла конфигурации.
2. Указать новую версию образа в файле конфигурации, загрузить образы: docker compose -f docker-compose.prod.yml pull .
3. Перезапустить службы: docker compose -f docker-compose.prod.yml up -d .
4. Применить миграции базы данных: alembic upgrade head .
5. Выполнить проверку по разделу 8.

Откат выполняется указанием предыдущей версии образа и повторным запуском служб; при необходимости откатывается схема базы данных командой alembic downgrade <версия> либо восстанавливается резервная копия. Сообщения, находящиеся в обработке на момент обновления, не теряются: их состояние сохранено в базе данных, обработка возобновляется после запуска служб.

10. Резервное копирование и восстановление

- База данных PostgreSQL — регулярное резервное копирование средствами pg_dump или средствами резервного копирования заказчика. Основной и обязательный объект резервного копирования: в базе хранятся сообщения, правила, задачи и снимки состояния конвейера.
- Вложения (S3-совместимое хранилище) — штатные механизмы репликации или выгрузки объекта.
- Векторное хранилище Qdrant — штатный механизм снимков (snapshots). Допускается не резервировать: векторный индекс восстанавливается повторным построением по истории сообщений.
- Очередь NATS JetStream — допускается не резервировать: необработанные сообщения повторно получаются из почтовых ящиков при следующем опросе.
- Файл конфигурации .env — хранение в защищённом хранилище секретов заказчика.

11. Типовые ошибки установки и их устранение

Признак	Причина и способ устранения
Контейнер службы перезапускается, в журнале — ошибка подключения к базе данных	Неверная строка подключения либо недоступность СУБД. Проверить параметры подключения и сетевую доступность порта 5432
Служба запущена, письма не появляются	Неверные учётные данные ящика либо недоступность почтового сервера. Проверить протокол, порт и учётные данные в настройках ящика; для EWS — доступность службы веб-служб Exchange
Письма появляются, обработка не выполняется	Недоступен брокер сообщений NATS либо упал контейнер обработки. Проверить порт 14222 и журналы контейнера worker; после перезапуска обработка возобновится с прерванного шага
Письмам не присваиваются категории, извлечение данных не выполняется	Недоступен шлюз языковых моделей либо неверный ключ доступа. Проверить адрес шлюза и ключ; при недоступности модели конвейер работает в детерминированном резервном режиме по правилам
Вход в веб-интерфейс завершается ошибкой	Неверно настроен клиент OIDC: несовпадение адреса перенаправления, идентификатора или секрета клиента
Дашборд не обновляется в реальном времени	Обратный прокси не передаёт соединения WebSocket. Включить проксирование заголовков Upgrade и Connection для пути WebSocket
Аварийное завершение службы по превышению памяти	Недостаточный объём оперативной памяти. Увеличить объём памяти либо уменьшить лимиты выборки и параллельность обработки

12. Ограничения

- Работа в изолированном контуре без доступа к сети «Интернет» возможна при условии размещения языковой модели в контуре заказчика; коннекторы к облачным внешним сервисам (мессенджеры, облачные CRM) в этом режиме недоступны — остаются почтовые протоколы, внутренние коннекторы и универсальный HTTP-коннектор.
- Качество классификации и маршрутизации зависит от настройки категорий, правил и матрицы маршрутизации под процессы заказчика.
- Число одновременно обрабатываемых сообщений ограничивается вычислительными ресурсами сервера и лимитами шлюза языковых моделей; при потоке свыше 10 000 сообщений в сутки рекомендуется увеличение ресурсов серверной части.

13. Техническая поддержка

ООО «СЭНДБОКСЕР», 107113, г. Москва, ул. 3-я Рыбинская, д. 18, стр. 2.
Электронная почта: contact@sandboxer.ru · Телефон: +7 (917) 240-02-43 · Сайт: sandboxer.ru

Порядок обращения в техническую поддержку, классификация обращений и сроки реакции приведены в документе «Описание процессов, обеспечивающих поддержание жизненного

цикла программного обеспечения».

© 2026 ООО «СЭНДБОКСЕР». Все права защищены. Документ является эксплуатационной документацией на программное обеспечение «TaskFlow AI».