

Инструкция по установке программного обеспечения «ITQ CRM»

Аннотация

Настоящий документ является инструкцией по установке (развёртыванию) программного обеспечения «ITQ CRM» (далее — Программное обеспечение, ПО, Система) и предназначен для системных администраторов, инженеров эксплуатации.

ПО «ITQ CRM» представляет собой корпоративную систему управления взаимоотношениями с клиентами (CRM) для организаций корпоративного сектора: управление корпоративными клиентами (юридическими лицами), партнёрами, сделками, продажами, задачами и мультиканальными коммуникациями (телефония, электронная почта, мессенджеры). Система реализована как клиент-серверное веб-приложение, доступное через веб-браузер.

ПО «ITQ CRM» поставляется в двух вариантах: (1) по сервисной (SaaS) модели — единый централизованный экземпляр на серверах правообладателя на территории РФ, доступ через веб-браузер по HTTPS; (2) в варианте локальной установки (on-premise) — развёртывание в собственной инфраструктуре заказчика на его вычислительных мощностях на территории РФ, в том числе в закрытом контуре без доступа в Интернет. В обоих вариантах используется единый дистрибутив (набор контейнеров Docker); механизм лицензионных ключей не применяется.

Технологический стек Системы:

- серверная часть (бэкенд): FastAPI, SQLAlchemy 2 (асинхронный режим), Python 3.11; управление миграциями базы данных — Alembic; планировщик фоновых задач — APScheduler;
- клиентская часть (фронтенд): React 18, TypeScript, сборщик Vite, CSS-фреймворк Tailwind;
- система управления базами данных — PostgreSQL 16; кэш и хранилище лимитера запросов — Redis 8;
- контейнеризация — Docker; обратный прокси и терминатор HTTPS — Nginx; продуктивное развёртывание выполняется по схеме «сине-зелёного» обновления без простоя (blue-green).

Установка Системы выполняется путём развёртывания набора контейнеров Docker; отдельный бинарный инсталлятор не применяется, механизм лицензионных ключей в изделии не используется. Обязательными инфраструктурными компонентами являются только PostgreSQL и Redis, разворачиваемые как контейнеры из состава поставки. Все внешние интеграции (реализованные на российских сервисах) являются опциональными: при незадаанных ключах соответствующие функции автоматически отключаются, при этом установка и запуск Системы выполняются успешно.

Опциональные функции искусственного интеллекта (формирование рекомендаций «следующего лучшего действия» / «следующего лучшего предложения», NBA/NBO, на базе российской языковой модели GigaChat — большой языковой модели ПАО «Сбербанк», размещённой на территории Российской Федерации) не входят в число обязательных зависимостей и активируются исключительно при установке признака ``ai_features_enabled`` и наличии ключа доступа к сервису GigaChat; GigaChat доступен напрямую с территории РФ, иностранные сервисы и прокси не требуются. Без указанных условий Система функционирует в полном объёме за исключением функций искусственного интеллекта.

Исходный текст ПО хранится в системе контроля версий Git в защищённом репозитории, размещённом на собственной инфраструктуре правообладателя на территории Российской Федерации по адресу 196006, г. Санкт-Петербург, ул. Цветочная, д.21, лит. А. Объектный код (собранные Docker-образы) и развёрнутый экземпляр ПО размещаются на арендованном сервере российского провайдера облачной инфраструктуры Timeweb Cloud; серверные мощности и дата-центр расположены на территории Российской Федерации.

Документ подготовлен в соответствии с требованиями к сведениям, представляемым для включения в Единый реестр российских программ для электронных вычислительных машин и баз данных.

1. Системные требования

1.1. Требования к серверной части (хост развёртывания)

Операционная система и системное программное обеспечение:

- операционная система — дистрибутив Linux с поддержкой контейнеризации Docker; штатной продуктивной средой является Ubuntu 22.04 LTS (архитектура процессора x86_64/amd64);
- Docker Engine и Docker Compose версии 2 (синтаксис ``docker compose``);
- Nginx на хосте — для продуктивного развёртывания (терминация HTTPS и переключение слотов blue-green);
- Node.js и менеджер пакетов npm — на машине сборки (для формирования статической сборки фронтенда).

Требования к вычислительным ресурсам определены исходя из лимитов контейнеров Системы:

- среда разработки (dev) — суммарно ориентировочно 2 виртуальных процессора и около 2 ГБ оперативной памяти (контейнер базы данных — 0,5 vCPU / 512 МБ; Redis — 0,5 vCPU / 256 МБ; сервер приложений — 1 vCPU / 1 ГБ; фронтенд — 0,5 vCPU / 256 МБ);
- продуктивная среда (blue-green): контейнер базы данных — 1 vCPU / 1 ГБ; Redis — 0,5 vCPU / 256 МБ (ограничение памяти 200 МБ); каждый слот сервера

приложений — до 2 ГБ памяти при резервировании 512 МБ, 1 vCPU, сервер приложений uvicorn с двумя рабочими процессами;

- рекомендуемый минимум продуктивного хоста с учётом одновременного существования двух слотов в момент переключения — не менее 4 виртуальных процессоров и 8 ГБ оперативной памяти (фактический продуктивный сервер — 8 vCPU / 12 ГБ ОЗУ / около 97 ГБ дискового пространства, используется совместно с иными проектами).

Требования к дисковому пространству:

- том базы данных, том Redis (append-only file), каталог статических файлов (порядка 196 МБ), каталог резервных копий (порядка 200 МБ в сутки), образы Python с компонентами оптического распознавания (несколько ГБ);
- рекомендуемый минимальный объём диска — от 40 ГБ; для продуктивной нагрузки — 80–100 ГБ.

1.2. Требования к клиентской части (рабочее место пользователя)

- Тонкий клиент — современный веб-браузер; установка прикладного программного обеспечения на рабочее место не требуется.
- Операционная система рабочего места — любая (Windows, macOS, Linux).
- Поддерживаются актуальные версии браузеров Яндекс Браузер, Google Chrome, Microsoft Edge, Mozilla Firefox, Apple Safari с поддержкой стандарта ECMAScript-модулей.
- Для функций программной телефонии (WebRTC-софтфон) необходимы доступ к микрофону и защищённое соединение HTTPS (политика браузера в отношении WebRTC).
- Сетевой доступ к продуктивному адресу Системы по протоколу HTTPS.

1.3. Требования к внешним зависимостям

PostgreSQL и Redis разворачиваются как контейнеры из состава поставки (отдельная установка не требуется); схемы и расширения базы данных создаются автоматически при инициализации, часть схем — средствами миграций Alembic.

Все внешние интеграции (российские почтовые сервисы Яндекс 360, Mail.ru для бизнеса по протоколам IMAP/SMTP; российское облачное хранилище Яндекс Диск; календарь Яндекс Календарь; мессенджеры MAX и Telegram, в том числе через российский агрегатор Wazzup24; телефония Exolve VATS (МТС); HH.ru; сервис рассылок UniSender Go; приём входящих сообщений по IMAP; российский сервис искусственного интеллекта GigaChat) являются опциональными: при незаданных ключах соответствующие функции отключаются, установка успешна. Функции искусственного интеллекта используют российский сервис GigaChat, доступный напрямую с территории Российской Федерации (иностранные сервисы и прокси не требуются); при незаданном ключе доступа к GigaChat функции искусственного интеллекта неработоспособны, при этом остальная функциональность Системы сохраняется в полном объёме.

2. Процесс установки для экспертной проверки (доступ к проверочному экземпляру)

Для проверки функциональных характеристик Системы предоставляется доступ к предварительно развёрнутому проверочному (демонстрационному) экземпляру ПО «ITQ CRM». Установка программного обеспечения на стороне эксперта не требуется — доступ осуществляется через веб-браузер.

2.1. Точки доступа к проверочному экземпляру

- Рабочий веб-интерфейс Системы: `https://crm.itquick.ru/` — страница входа и защищённая рабочая область.
- Проверка работоспособности серверной части (программный интерфейс): `https://crm.itquick.ru/api/v1/health` — возвращает статус готовности с проверкой доступности PostgreSQL и Redis (`{"status": "ok", ...}`; при недоступности хранилищ — код ответа 503).
- Интерактивная документация программного интерфейса (Swagger UI — `/api/docs`, ReDoc — `/api/redoc`, схема OpenAPI) доступна только при включённом признаке отладки (`APP_DEBUG=true`); на продуктивном стенде документация отключена.

2.2. Аутентификация эксперта

- Вход выполняется по учётным данным (адрес электронной почты и пароль); по результату аутентификации выдаётся токен доступа JWT.
- Учётные данные для эксперта Реестра (логин и пароль проверочной учётной записи с необходимым набором прав) предоставляются правообладателем отдельно и в настоящем документе не приводятся [УТОЧНИТЬ: логин/пароль проверочной учётной записи; при необходимости — тестовая учётная запись администратора].

2.3. Предоставление удалённого доступа (при необходимости развёртывания у эксперта)

В случае если проверка предусматривает самостоятельное развёртывание Системы силами эксперта, правообладателем предоставляются:

- доступ к репозиторию исходного кода (по разделу 3 настоящей инструкции);
- при необходимости — удалённый доступ (SSH) к выделенному стенду для наблюдения за процессом развёртывания.

3. Процесс установки (развёртывание Системы)

Раздел описывает полный цикл развёртывания Системы из исходного кода. Приведён порядок для среды разработки/локального запуска (пункт 3.2) и для продуктивной среды по схеме blue-green (пункт 3.3).

3.1. Предварительные условия

- На хосте установлены Docker Engine и Docker Compose версии 2.

- Получен доступ к репозиторию исходного кода. Исходный код размещается в самостоятельно размещаемом (self-hosted) экземпляре GitLab на территории Российской Федерации по адресу г. Санкт-Петербург, ул. Цветочная, д.21, лит. А.
- Для продуктивной среды на хосте установлен и настроен Nginx; для сборки фронтенда доступны Node.js и npm.

3.2. Развёртывание в среде разработки (локальный запуск)

Шаг 1. Получение исходного кода.

```
git clone <адрес-репозитория> itq_crm
```

```
cd itq_crm
```

Шаг 2. Конфигурация переменных окружения серверной части (файл `.env`).

```
cp backend/.env.example backend/.env
```

В файле `backend/.env` заполняются как минимум следующие параметры:

- ``APP_SECRET_KEY`` — секрет приложения (генерация: ``openssl rand -hex 32``);
- ``JWT_SECRET_KEY`` — секрет для подписи токенов JWT (генерация: ``openssl rand -hex 32``);
- ``CRM_DB_PASSWORD`` — пароль базы данных PostgreSQL;
- ``REDIS_PASSWORD`` — пароль Redis (в среде разработки допускается пустое значение);
- ``SEED_DEFAULT_PASSWORD`` — пароль для создаваемых учётных записей;
- ``SEED_ADMIN_EMAIL`` — адрес администратора;
- ``WEBHOOK_API_KEY`` — ключ приёма входящих заявок;
- при необходимости — ``OAUTH_CLIENT_ID``, ``OAUTH_CLIENT_SECRET`` (единый вход Яндекс ID / VK ID) и иные параметры опциональных интеграций.

Секреты в состав сведений для Реестра не выносятся и в репозиторий не помещаются.

Шаг 3 (опционально). Конфигурация переменных окружения клиентской части.

```
cp frontend/.env.example frontend/.env.local
```

Шаг 4. Сборка и запуск контейнеров.

```
docker compose build
```

```
docker compose up -d
```

Сервер приложений запускается после перехода контейнеров базы данных и Redis в состояние готовности (healthy).

Шаг 5. Применение миграций базы данных (Alembic).

```
docker compose exec crm-api alembic upgrade head
```

Миграции применяются вручную: сценарий запуска контейнера (`entrypoint.sh`) миграции не выполняет — он лишь настраивает права доступа к каталогам и передаёт управление серверу приложений. Миграции, помимо основных схем, создают схемы ``chat``, ``email``, ``telegram``.

Шаг 6. Первичная настройка: наполнение справочников и создание администратора.

```
docker compose exec crm-api python -m app.utils.seed_data
```

```
docker compose exec crm-api python create_admin.py
```

Наполнение справочников и учётных записей выполняется после применения миграций; создание администратора требует заданного значения ``SEED_DEFAULT_PASSWORD`` и наличия роли ``ADMIN``.

Шаг 7. Проверка работоспособности.

```
docker compose ps
```

```
curl http://localhost:8000/api/v1/health
```

Ожидаемый результат проверки готовности — ``{"status":"ok", ...}``.

Точки доступа в среде разработки: веб-интерфейс — ``http://localhost:3000`` (dev-сервер Vite) либо ``http://localhost:80`` (контейнер фронтенда); документация программного интерфейса — ``http://localhost:8000/api/docs`` (только при ``APP_DEBUG=true``).

3.3. Развёртывание в продуктивной среде (blue-green, обновление без простоя)

Предварительные условия на сервере: каталог проекта ``/opt/itq_crm``; сопровождаемый вручную файл ``backend/.env`` (из синхронизации исключён); установленные Docker Compose v2 и Nginx; созданные внешняя сеть и внешние тома данных.

Шаг 1. Конфигурация параметров развёртывания.

```
cp deploy.env.example deploy.env
```

Заполняются параметры ``DEPLOY_SERVER`` (адрес продуктивного сервера) и ``DEPLOY_REMOTE_DIR`` (``/opt/itq_crm``).

Шаг 2. Запуск развёртывания с рабочей машины.

```
./deploy.sh
```

Шаг 3. Автоматизированный цикл на сервере. Серверный сценарий развёртывания выполняет:

- поднятие общих сервисов (база данных, Redis);
- сборку нового слота при работающем текущем слоте;
- предмиграционное резервное копирование базы данных (`pg_dump | gzip``); при неуспехе резервного копирования развёртывание прерывается;
- применение миграций базы данных (`alembic upgrade head``);
- проверку готовности нового слота (до 60 секунд);
- переключение обратного прокси Nginx на новый слот (с проверкой конфигурации и перезагрузкой; при ошибке — автоматический откат);
- фиксацию активного слота, остановку прежнего слота и очистку неиспользуемых образов.

Шаг 4. Откат обновления (при необходимости).

```
ssh <DEPLOY_SERVER> "bash /opt/itq_crm/deploy/rollback.sh"
```

Слоты blue/green публикуются на портах 3081 и 3082 соответственно; наружу открывается только активный слот фронтенда, внешний Nginx хоста завершает HTTPS и проксирует запросы на активный слот (продуктивный адрес — `https://crm.itquick.ru``).

Приведённый порядок развёртывания применим как к размещению по модели SaaS на серверах правообладателя, так и к локальной установке (on-premise) в инфраструктуре заказчика на его вычислительных мощностях на территории РФ, в том числе в закрытом контуре: заказчику передаётся единый дистрибутив (набор контейнерных образов Docker, файлы `docker compose`` и сопроводительная документация), развёртывание выполняется теми же средствами.

3.4. Основные переменные окружения

Конфигурация читается из файла `backend/.env``. Ключевые параметры серверной части:

- `APP_ENV`` / `APP_DEBUG`` — окружение и режим отладки (влияет на доступность API-документации); по умолчанию `development`` / `true``;
- `APP_SECRET_KEY`` — секрет приложения (обязателен);
- `CRM_DB_HOST`` / `CRM_DB_PORT`` / `CRM_DB_NAME`` / `CRM_DB_USER`` — параметры PostgreSQL; `CRM_DB_PASSWORD`` — пароль (обязателен);
- `CRM_DB_POOL_SIZE`` / `CRM_DB_MAX_OVERFLOW`` — пул соединений с базой данных;
- `REDIS_HOST`` / `REDIS_PORT`` / `REDIS_DB`` / `REDIS_PASSWORD`` / `REDIS_PREFIX`` — параметры Redis;
- `JWT_SECRET_KEY`` — секрет подписи JWT (обязателен); `JWT_ALGORITHM`` (по умолчанию `HS256``), время жизни токена доступа (по умолчанию 60 минут) и токена обновления (по умолчанию 30 дней);

- ``CORS_ORIGINS`` — разрешённые источники запросов;
- ``IMPORT_MAX_FILE_SIZE_MB`` / ``IMPORT_ALLOWED_EXTENSIONS`` — ограничения импорта данных (по умолчанию 50 МБ; форматы ``xlsx,xls,csv``);
- параметры опциональных интеграций (``OAUTH_*``, ``INTAKE_IMAP_*`` / ``INTAKE_ENABLED``, ``UNISENDER_*``, ``HH_*``, ``EXOLVE_*``, ``WAZZUP_*``, ``TELEGRAM_*``, ``WEBHOOK_API_KEY``) — по умолчанию не заданы (функции отключены);
- ``GIGACHAT_API_KEY`` (ключ авторизации российского сервиса ИИ GigaChat) / ``GIGACHAT_SCOPE`` / ``LLM_DEFAULT_MODEL`` (по умолчанию ``GigaChat``) — параметры опциональных функций искусственного интеллекта; при незаданном ключе функции искусственного интеллекта отключены; GigaChat доступен напрямую с территории Российской Федерации, внешний прокси не требуется;
- ``SEED_DEFAULT_PASSWORD`` / ``SEED_ADMIN_EMAIL`` — параметры первичной инициализации учётных записей;
- ``LOG_LEVEL`` / ``LOG_FORMAT`` — параметры журналирования (по умолчанию ``INFO`` / ``json``).

В продуктивной конфигурации параметры ``CRM_DB_PASSWORD`` и ``REDIS_PASSWORD`` являются обязательными: при их отсутствии запуск контейнеров прерывается. Секреты в состав сведений для Реестра не выносятся.

3.5. Используемые сетевые порты

- PostgreSQL — ``5433`` → `5432`` (в среде разработки; в продуктивной среде наружу не публикуется);
- Redis — ``6380`` → `6379`` (в среде разработки; в продуктивной среде наружу не публикуется);
- сервер приложений (uvicorn) — ``8000`` → `8000`` (в продуктивной среде доступ через Nginx);
- фронтенд (nginx) — ``80`` → `80`` (среда разработки); dev-сервер Vite — ``3000``;
- фронтенд продуктивной среды — ``3081`` → `80`` (blue) и ``3082`` → `80`` (green);
- программная телефония (SIP/RTP, опционально) — ``5060/udp``, ``10000-10100/udp``;
- внешний Nginx хоста — ``443`` / ``80`` (продуктивная среда).

3.6. Проверка после установки и штатное обновление

- Контроль состояния: ``docker compose ps`` (все сервисы в состоянии healthy); проверка готовности ``curl .../api/v1/health`` (``{"status":"ok"}``; код 503 — при недоступности базы данных или Redis); контроль импорта модуля приложения ``docker compose exec crm-api python -c "import app.main"``.
- Ручное резервное копирование базы данных: ``docker compose exec crm-db pg_dump -U crm_user itq_crm > backup.sql`` (в цикле продуктивного развёртывания резервная копия создаётся автоматически перед применением миграций).
- Штатное обновление в среде разработки: ``git pull`` → ``docker compose up --build -d crm-api crm-frontend`` → ``docker compose exec crm-api alembic upgrade head``.

- Штатное обновление в продуктивной среде: запуск `./deploy.sh` (цикл blue-green по пункту 3.3).`

4. Информация для контактов

Правообладатель и разработчик: Общество с ограниченной ответственностью «АЙТИ ХАЙТЭК».

ИНН 7743355630, КПП 774301001, ОГРН 1217700097120.

Адрес: 125315, г. Москва, вн. тер. г. муниципальный округ Сокол, пр-кт Ленинградский, д. 80Б, к. 3, помещ. 14/Т.

Телефон: +7 (495) 147-37-06.

Адрес электронной почты: i@itquick.ru.