

Установка и эксплуатация

Программа: «Сокол — система автоматизированного анализа заданий на проектирование».

Сокол предоставляется пользователям как веб-сервис. Рабочее место пользователя и серверная часть имеют разные требования. Пользователь не устанавливает серверные компоненты и не получает локальный установщик.

Рабочее место пользователя

Для работы нужны доступ к сайту по HTTPS, браузер с включённым JavaScript и возможностью загрузки файлов, а также программа для чтения результатов DOCX. Для входа требуются учётные данные, выданные правообладателем. Поддерживаемые входные файлы для штатного сценария — PDF и DOCX размером до 50 МБ.

Минимальные значения для процессора, памяти, диска и перечень проверенных версий браузеров для рабочего места не установлены; фиксированные числовые требования здесь не заявляются. Установка Сокола на компьютер пользователя не требуется.

Серверная часть

Развёртывание выполняет оператор сервиса в контролируемой им инфраструктуре из исходного кода и конфигурации поставки. Фактическая схема содержит четыре компонента: веб-интерфейс на React, API на Python/FastAPI, фоновый обработчик задач Celery и очередь Redis. Веб-интерфейс собирается в статические файлы и отдаётся Nginx; внешний обратный прокси предоставляет HTTPS-доступ.

Поставка исходного кода содержит `backend/Dockerfile`, `frontend/Dockerfile` и `docker-compose.yml`. В конфигурации используются образы Python 3.11, Node 20 для сборки, Nginx 1.27 и Redis 7. API и обработчик задач используют общий образ `backend`. Каталоги `backend/data`, `backend/uploads` и `backend/output` подключаются к API и обработчику как постоянные тома. Для анализа требуется подключение к используемому сервисом LLM API; его параметры доступа хранятся в приватной конфигурации и не публикуются.

Порядок развёртывания оператором

1. Подготовить сервер с Docker Engine, Docker Compose и обратным прокси для HTTPS. Конкретные минимальные аппаратные ресурсы для серверной части не зафиксированы; их определяют по нагрузке и объёму документов.
2. Разместить проверенную версию исходного кода и приватную конфигурацию сервиса, включая параметры подключения к LLM API. Создать или сохранить постоянные каталоги данных, загрузок и результатов. Не включать учётные данные и ключи внешних сервисов в публичные файлы.
3. Собрать образ `backend` командой `docker build -t sokol-backend ./backend`, затем образ `frontend` командой `docker compose build frontend` из корня поставки.
4. Запустить компоненты через Docker Compose. Настроить HTTPS-прокси на веб-интерфейс и убедиться, что API доступен через маршрут `/api/`.
5. Проверить состояние API, Redis и фонового обработчика, затем вход под специально выделенной проверочной учётной записью. Работоспособность API сама по себе не

подтверждает обработку документов: состояние фонового обработчика проверяется отдельно.

Оператор должен сохранять резервные копии постоянных данных и конфигурации, контролировать свободное место на сервере и состояние очереди/обработчика. Перед обновлением следует зафиксировать используемую версию, сохранить данные и обеспечить возможность возврата к прежнему образу. Смена версии выполняется в согласованное эксплуатационное окно.