УТВЕРЖДЁН
RU.EDZS.00001-01 32 01-ЛУ

Сервис публикации результатов оценочных процедур, ответов, бланков ответов, подачи апелляций, регистрации участников оценочных процедур

Руководство системного программиста. Инструкция по установке

RU.EDZS.00001-01 32 01

Редакция 6 от 05.11.2026

Листов 10

Екатеринбург 2026

Подп. и дата
Инв. № дубл.
Взам. инв. №
Подп. и дата
Инв. № подл.

АННОТАЦИЯ

Документ относится к программе «Сервис публикации результатов оценочных процедур, ответов, бланков ответов, подачи апелляций, регистрации участников оценочных процедур». Установка, настройка, запуск, проверка, обновление и диагностика серверной части.

Документ относится к веб-поставке 2026.11.04-r3 и состоит из одной части.

СОДЕРЖАНИЕ

1 Общие сведения о программе 4
1.1 Требования к серверу 4
2 Структура программы 4
2.1 Минимальная архитектура 4
3 Настройка программы 5
3.1 Подготовка 5
3.2 Установка релиза 5
3.3 Параметры конфигурации 6
3.4 Первый администратор и доступ 6
3.5 Проверочное развёртывание 7
4 Проверка программы 7
4.1 Первичная проверка 7
5 Дополнительные возможности 8
5.1 Обновление и откат 8
5.2 Удаление 8
6 Сообщения системному программисту 8
7 Ссылочные документы и сокращения 9
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 10

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Руководство предназначено техническому администратору Linux. Для развёртывания требуются навыки Docker Compose, PostgreSQL, управления доступом и резервного копирования. Экспертный доступ внутри контейнера не даёт права изменять конфигурацию сервера.

1.1 Требования к серверу

- 64-разрядный Linux-сервер с Docker Engine и Docker Compose;
- контейнер приложения: Ruby 3.2.2, Rails 7.0.8 и Bundler 2.5.23;
- системные зависимости из Dockerfile, включая Node.js, libpq, libvips и Poppler;
- PostgreSQL 15;
- Redis 7.2;
- Nginx с HTTPS перед контейнером web; Puma запускается внутри web;
- достаточный объем диска с учетом числа и размера бланков.

Проверенное окружение: Ubuntu 24.04.4 на хосте, Debian 12 в контейнере; Ruby 3.2.2, Rails 7.0.8, Puma 6.4.0, PostgreSQL 15, Redis 7.2, Sidekiq 6.5.5. Российские ОС указаны как целевые, результаты испытаний на них отдельно не оформлены.

2 СТРУКТУРА ПРОГРАММЫ

2.1 Минимальная архитектура

Узел	Назначение
Веб-сервер приложения	Ruby on Rails и Puma; Nginx как обратный прокси.
Сервер базы данных	PostgreSQL; может быть совмещен с веб-сервером для тестового стенда.
Redis	Очереди фоновых заданий и служебные данные.
Хранилище файлов	Бланки, импортируемые файлы, экспорт и журналы с контролем доступа.

Браузер отправляет HTTPS-запрос; Nginx передаёт его Puma/Rails. Контроллер определяет сеанс и роль, проверяет доступ к записи и выполняет разрешённое действие через модель. Active Record обращается к PostgreSQL; загруженные файлы хранятся отдельно от строк БД. Представление формирует HTML, JavaScript и CSS. Длительные операции передаются Sidekiq через Redis; worker обрабатывает задания вне веб-запроса.

Для показа результата выбираются записи, доступные участнику; для технической апелляции дополнительно проверяются настройка экзамена и окно подачи. Сохраняются номер задания, распознанное и предложенное значения, комментарий и статус. Решение оператора выполняется отдельным действием с проверкой полномочий.

3 НАСТРОЙКА ПРОГРАММЫ

3.1 Подготовка

1. создать отдельного системного пользователя приложения;
2. подготовить DNS-имя и сертификат TLS;
3. подготовить POSTGRES_USER, POSTGRES_PASSWORD и POSTGRES_DB для создания новой БД при первом запуске;
4. подготовить каталог полного комплекта развёртывания и закрытое хранилище резервных копий;
5. получить дистрибутив от правообладателя и проверить контрольную сумму;
6. подготовить значения секретов и переменных окружения.

3.2 Установка релиза

Команды выполняются на хосте из каталога полного комплекта поставки. До подготовки БД задайте параметры .env по подразделу 3.3 и данные первого администратора по подразделу 3.4. Для новой установки заранее измените name проекта, имя тома assets и ссылку на образ worker в compose.yml: она должна указывать на образ web этого проекта вместо lks-demo-web. Затем выполните команды последовательно:

1) docker compose config --quiet — проверка синтаксиса Compose и доступности файла окружения без вывода секретов.

2) docker compose build — сборка образа приложения. Дождитесь кода завершения 0; при ошибке сохраните журнал и не переходите к подготовке базы.

3) docker compose up -d db redis — запуск PostgreSQL и Redis.

4) docker compose exec -T db sh -c 'pg_isready -U "$POSTGRES_USER" -d "$POSTGRES_DB"' — проверка готовности базы. При отказе дождитесь запуска и проверьте журнал db.

5) docker compose run --rm --no-deps web bundle exec rails db:prepare — создание схемы новой учебной базы и запуск seeds. Для существующей базы сначала создайте резервную копию; повторная подготовка не является способом восстановления данных.

6) docker compose run --rm --no-deps -e SKIP_JS_BUILD=1 web bundle exec rails assets:precompile — сборка клиентских ресурсов.

7) docker compose up -d web worker — запуск веб-приложения и фонового обработчика.

8) docker compose ps; docker compose logs --tail 100 web worker — проверка состояния и журналов. Затем выполните контроль входа, результата, PDF и ролей по разделу 4.

Настройте DNS и обратный прокси для выбранного домена: запросы к приложению передаются на 127.0.0.1:18767, который Compose связывает с портом 3000 контейнера web. Внешний доступ должен использовать HTTPS. Порт приложения оставьте привязанным к localhost; после настройки проверьте вход и ссылки через внешний адрес.

3.3 Параметры конфигурации

В поставке Docker файл database.yml копируется в config/database.yml и читает DATABASE_URL. Параметры прежнего развёртывания LKS_DATABASE_* к этой поставке не относятся. Значения окружения задаются в .env рядом с compose.yml и передаются сервисам через env_file.

Параметры	Назначение и требования
DATABASE_URL	Обязателен для web и worker. PostgreSQL URL: postgresql://USER:PASSWORD@db:5432/DB. USER, PASSWORD и DB должны совпадать с POSTGRES_*; специальные символы пароля кодируют по правилам URL.
POSTGRES_USER, POSTGRES_PASSWORD, POSTGRES_DB	Обязательны для инициализации нового тома PostgreSQL. Непустые строки. Изменение .env не меняет пароль или название уже созданной базы.
REDIS_URL, REDIS_CACHE_URL	REDIS_URL — Redis для очередей; REDIS_CACHE_URL — для лимитов входа. Пример внутри Compose: redis://redis:6379/0 и redis://redis:6379/2. При отсутствии REDIS_CACHE_URL ограничитель использует REDIS_URL.
SECRET_KEY_BASE	Секрет Rails для web и worker. Непустая случайная строка; значение совпадает у сервисов одного экземпляра и хранится вне исходников.
BOOTSTRAP_ADMIN_EMAIL, BOOTSTRAP_ADMIN_LOGIN, BOOTSTRAP_ADMIN_PASSWORD	Для первоначального создания администратора через seeds. Существующую запись с той же почтой seeds не изменяет.
RAILS_ENV, RAILS_SERVE_STATIC_FILES, RAILS_LOG_TO_STDOUT	Dockerfile задаёт production, выдачу статики и журналирование в stdout. Параметры переопределяют только при согласованном изменении поставки.

3.4 Первый администратор и доступ

Для первого администратора задайте в закрытом .env BOOTSTRAP_ADMIN_EMAIL, BOOTSTRAP_ADMIN_LOGIN и BOOTSTRAP_ADMIN_PASSWORD. Подготовка новой БД запускает db/seeds.rb: при отсутствии пользователя с указанной почтой создаётся запись с admin=true. Если такая почта уже существует, seeds не меняет её пароль и роль.

Для уже подготовленной базы выполните docker compose run --rm --no-deps web bundle exec rails db:seed. Проверьте вход и доступ к /admin. После первичной настройки удалите BOOTSTRAP_ADMIN_PASSWORD из .env и пересоздайте web и worker командой docker compose up -d --force-recreate web worker, чтобы секрет не оставался в их окружении. Сохраните пароль в защищённом хранилище.

В данной поставке отдельный второй фактор не настроен. Арифметическая CAPTCHA ограничивает автоматические попытки входа и не заменяет второй фактор. Изменение пароля или роли выполняет уполномоченный администратор; доступ проверяется в новом сеансе.

3.5 Проверочное развёртывание

Полный комплект развёртывания содержит source.tar.gz, Dockerfile, compose.yml, database.yml, demo.rb и каталог overrides одного выпуска. Контрольная сумма source.tar.gz: 6c5eb5456ed555882688c472ab3d43905f517e0891eb8036ee804d79e3c94dbc. Одного исходного архива недостаточно для повторения сборки.

В .env задайте параметры из таблицы конфигурации. Команда docker compose config --quiet проверяет синтаксис конфигурации; корректность адресов и секретов проверяется запуском сервисов. Для отдельной установки имена проекта, томов и образов согласуют до запуска.

4 ПРОВЕРКА ПРОГРАММЫ

4.1 Первичная проверка

- главная страница открывается по HTTPS;
- администратор может войти;
- импорт тестового набора завершается без ошибок;
- участник видит тестовый результат и обезличенный бланк;
- тестовая апелляция регистрируется и видна оператору;
- фоновые задания исполняются;
- резервная копия создается и контрольное восстановление описано.

Проверка	Метод	Ожидаемый результат
Состояние служб	docker compose ps; просмотр журнала web и worker	Службы запущены, нет циклических перезапусков.
Вход	Открыть /users/sign_in, решить CAPTCHA и войти учебной учётной записью	Открыт кабинет; без ответа CAPTCHA вход отклоняется.
Материалы	Открыть учебный результат и бланк	Доступны результаты и PDF из трёх страниц.
Роли	Открыть /admin под участником и администратором	Участнику 403; администратору панель 200.
Статика	Открыть страницу и проверить загрузку assets	Bootstrap и штатный Action Text загружаются с домена стенда.

Это критерии приёмки установленного экземпляра. Подтверждённые проверки и непроверенные сценарии разделены в документе 09. При несоответствии эксплуатацию новой редакции не начинают; сохраняют журнал и возвращают предыдущую согласованную конфигурацию.

5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

5.1 Обновление и откат

Перед обновлением создаются резервные копии базы данных, загружаемых файлов и конфигурации. Новый релиз разворачивается отдельно, выполняются миграции и проверка. Откат производится возвратом предыдущего релиза и, если миграции необратимы, восстановлением согласованной резервной копии.

5.2 Удаление

Удаление выполняется уполномоченным администратором после архивирования обязательных данных и журналов. Системные учетные записи, ключи доступа и секреты отзываются. Требования к срокам хранения данных определяет заказчик с учетом применимого законодательства.

Руководство описывает поставку Docker Compose с Puma. Прежние конфигурации Capistrano/Passenger не входят в приведённую процедуру. Windows-утилита подготовки бланков поставляется отдельно; сведения о ней приведены в документе 05.

6 СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

Диагностику выполняют из каталога развёртывания /opt/lks-demo с полномочиями технического администратора. Проверить docker compose ps, затем docker compose logs --tail 100 web worker. Секреты из .env и журналов не включать в обращение поддержки.

Признак в журнале или HTTP	Значение	Порядок действий
ActiveRecord::ConnectionNotEstablished / PG::ConnectionBad	Соединение с PostgreSQL не установлено.	Проверить состояние db, адрес, имя БД, пользователя и доступность сети; не удалять том БД.
Redis::BaseError (ошибка соединения или выполнения команды Redis)	Соединение с Redis не установлено.	Проверить redis и его параметры у web и worker; после восстановления проверить вход и очередь.
ActiveRecord::PendingMigrationError	Схема БД отстаёт от исходного кода.	Создать резервную копию, проверить версию и выполнить миграции выбранного релиза.
ActionController::InvalidAuthenticityToken	Проверка подлинности формы не пройдена.	Открыть новую форму; проверить HTTPS, cookie, прокси и согласованность секретов экземпляра.
HTTP 404 для /assets/…	Запрошенный ресурс отсутствует.	Для действующего ресурса сверить importmap и сборку assets. Для удалённого устаревшего ресурса 404 ожидаем.
HTTP 502 на внешнем веб-сервере	Прокси не получил корректный ответ приложения.	Проверить web, его журнал и привязку 127.0.0.1:18767; затем конфигурацию прокси.

Признак в журнале или HTTP	Значение	Порядок действий
HTTP 403 на административном маршруте	Отказ по роли пользователя.	Проверить вход и роль admin; участнику доступ запрещён штатно.

Текст после имени исключения зависит от драйвера и причины ошибки. При обращении в поддержку приложите сообщение и время события без паролей и ключей.

7 ССЫЛОЧНЫЕ ДОКУМЕНТЫ И СОКРАЩЕНИЯ

ГОСТ 19.101-2024; ГОСТ 19.503-79. ГОСТ 19.104-78; ГОСТ 19.105-78; ГОСТ 19.106-78. Связанные эксплуатационные документы — 02–09 комплекта редакции 6 от 05.11.2026.

БД — база данных; ОС — операционная система; СУБД — система управления базами данных; CAPTCHA — проверочное задание при входе; HTTP — протокол обмена веб-данными; HTTPS — HTTP с защитой соединения TLS; PDF — формат электронного документа; SHA-256 — алгоритм контрольной суммы; TLS — защита транспортного соединения.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Измен. листы	Замен. листы	Новые листы	Аннул. листы	Всего листов	№ документа	Входящий № и дата	Подпись	Дата