Руководство системного программиста

Личные кабинеты участников ВСОШ

Редакция от 4 ноября 2026 года

ООО «ЭдЭтСофтДизайн»
Генеральный директор
Никитин Сергей Васильевич

Документ 02
Екатеринбург · 2026

АННОТАЦИЯ

Настоящий документ описывает программу «Личные кабинеты участников ВСОШ», исходную поставку от 27 октября 2026 года. Содержит сведения о составе, установке, настройке, проверке, сопровождении и диагностике серверной установки. Предназначен для администратора серверного узла и приложения.

СОДЕРЖАНИЕ

1 общие сведения о программе 4
2 структура программы 5
3 настройка программы 6
3.1 Подготовка узла и поставки 6
3.2 Конфигурация и секреты 6
3.3 Сборка и запуск 7
3.4 Первоначальный административный доступ 7
4 проверка программы 9
4.1 Приёмочная проверка установки 9
4.2 Текущий контроль очередей 9
5 сопровождение и резервирование 10
5.1 Остановка и возобновление 10
5.2 Создание и проверка резервной копии 10
5.3 Аварийное восстановление 11
5.4 Обновление 11
6 дополнительные возможности 13
7 сообщения системному программисту 14

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Программа «Личные кабинеты участников ВСОШ» обеспечивает публикацию работ и результатов, предоставление критериев и разборов, приём апелляций и публикацию решений. Руководство предназначено для администратора серверной установки. Пользовательские операции описаны в отдельном руководстве.

Настоящая редакция относится к исходной поставке от 27 октября 2026 года. Перед выполнением команд администратор проверяет, что поставка содержит compose.yml, Dockerfile, .env.example, bin/backup_cycle, bin/restore_check и задачу operations:check. Примеры ниже выполняются на выделенном сервере из корневого каталога поставки. Команды docker compose запускаются на узле; команды после имени сервиса выполняются в контейнере.

Целевая серверная среда — РЕД ОС 8, архитектура amd64, Docker Engine с модулем Compose, SELinux в режиме Enforcing. Образы закреплены в поставке. Приложение использует Ruby 3.4.12, Rails 8.1.4 и PostgreSQL 16. Браузер подключается по HTTPS. Для сборки требуется доступ к реестрам образов и источникам зависимостей. После сборки основные функции не требуют внешнего сервиса распознавания.

Минимальные аппаратные характеристики и допустимый сезонный объём должны определяться испытанием конкретной установки. Измеренные пределы и гарантированное время восстановления в этой поставке не установлены. Размер диска рассчитывают с учётом оригиналов, производных PDF, базы, временной обработки и нескольких резервных копий; одно значение числа участников не определяет необходимый объём.

Администратор узла устанавливает контейнерную среду, управляет секретами, сетью, копиями и обновлением. Администратор приложения приглашает сотрудников и назначает права. Координатор управляет сезонами и турами. Эти полномочия различаются: роль в приложении не предоставляет доступа к оболочке сервера.

2 СТРУКТУРА ПРОГРАММЫ

Внешний запрос проходит через HTTPS-прокси к службе web. Web обращается к PostgreSQL и закрытому файловому тому files. Порт PostgreSQL наружу не публикуется; приложение слушает порт 3000 на loopback узла. Файлы работ не помещаются в public и выдаются после проверки прав.

Сервисы compose.yml: db — PostgreSQL; prepare — однократная подготовка схемы; web — браузерные запросы; worker — печать, сканы и публикации; registry — импорт и номера; scheduling — распределение заседаний; mail — отправка почты. Prepare должен завершиться успешно до старта зависимых служб. Имя процесса в исходном коде может отличаться: например, сервис registry запускает bin/registry_worker.

Профиль proxy добавляет HTTPS-прокси. Профиль ops предоставляет сервис backup для копирования и контрольного восстановления. Профили max и max-setup включаются отдельно. Их не следует добавлять к командам начальной установки без настройки интеграции.

Сохраняемые данные находятся в томах database_redos и files. Тома proxy_redos_data и proxy_redos_config содержат данные прокси. Файл .env и ключевой материал хранятся отдельно от исходников и от резервных копий. Удаление контейнера не означает удаления тома; команда down -v, напротив, удаляет тома и для обычного обслуживания не применяется.

3 НАСТРОЙКА ПРОГРАММЫ

3.1 Подготовка узла и поставки

До запуска администратор получает доверенную поставку, проверяет её контрольную сумму, подготавливает DNS-имя, доступ к портам 80/443 и синхронизацию времени. Исходный архив для указанного выпуска имеет SHA-256 73b1b3904eab4e577bdd0f14a310f8057b61f21dd0330441374d6f2c1c809c5d. Значение относится именно к архиву поставки, а не к произвольному архиву рабочей папки.

Проверьте установленную контейнерную среду и защиту узла:

```
docker version
docker compose version
getenforce
```

Ожидается успешное получение версий и ответ Enforcing. При отсутствии Docker или Compose сначала установите их по утверждённому регламенту площадки. В поставку не входит универсальный установщик контейнерного движка. Не продолжайте при выключенной защите, подменяя конфигурацию на privileged.

На выделенном узле РЕД ОС установите штатные шаблоны container-selinux/udica из дистрибутива. Из корня доверенной поставки выполните:

```
sudo python3 script/install_redos_sandbox.py
```

Установщик проверяет ОС, режим SELinux и происхождение файлов; устанавливает собственный модуль, не изменяя штатный container_t. Ошибка установки является основанием остановить развёртывание и разобрать причину. Успешная установка ещё не подтверждает изоляцию обработчиков: соответствующие проверки выполняются после запуска.

3.2 Конфигурация и секреты

Создайте закрытую копию образца окружения:

```
umask 077
cp .env.example .env
chmod 600 .env
```

Откройте .env в редакторе и замените все демонстрационные значения. PUBLIC_URL — внешний корневой HTTPS-адрес без нестандартного порта, например https://vsoshlk.example.org; PUBLIC_HOST — только имя vsoshlk.example.org. Эти адреса являются примерами и заменяются зарегистрированным именем установки. DNS должен указывать на сервер прокси.

SECRET_KEY_BASE — постоянный случайный секрет приложения. Создайте его отдельно, например openssl rand -hex 64, и перенесите в .env через защищённый редактор. POSTGRES_PASSWORD, POSTGRES_ADMIN_PASSWORD и RESTORE_DATABASE_PASSWORD создаются независимо; каждый должен содержать не менее 16 символов и отличаться от остальных. Для каждого можно использовать openssl rand -hex 32. Не выполняйте команды с секретами в общей записи терминала и не отправляйте содержимое .env в поддержку.

Назначение параметров: POSTGRES_PASSWORD получает рабочая роль vsosh; POSTGRES_ADMIN_PASSWORD используется только при администрировании кластера; RESTORE_DATABASE_PASSWORD — только отдельной ролью vsosh_restore для проверки копий. Веб-процесс не должен получать административные реквизиты или реквизиты восстановления. SECRET_KEY_BASE

требуется для проверки подписи и расшифрования резервной копии; утрата исходного значения делает копию непригодной для восстановления. При смене секрета сохраняйте его версии в отдельном защищённом хранилище.

WEB_CONCURRENCY=0 и RAILS_MAX_THREADS=3 — исходные настройки образца, а не гарантированные оптимальные значения. Изменяйте их по результатам испытаний и контролируйте число соединений PostgreSQL. MAX_ENABLED и внешние способы входа при первоначальной проверке оставьте выключенными.

Для почты укажите SMTP_ADDRESS, SMTP_PORT, SMTP_USERNAME, SMTP_PASSWORD, MAIL_FROM и MAIL_DOMAIN; APPEALS_EMAIL — адрес назначения служебных уведомлений. Значения получает сервис mail. Используйте согласованный SMTP-сервер с проверкой сертификата. До проверки реальной доставки не считайте сохранённое событие или успешную очередь подтверждением получения письма.

Перед сборкой проверьте синтаксис Compose без вывода развёрнутых секретов:

```
docker compose --profile proxy config --quiet
```

Ожидается код 0 без сообщения об отсутствующей обязательной переменной. При ошибке исправьте указанный параметр и повторите проверку. Не включайте полный вывод docker compose config в открытые журналы: он может содержать значения секретов.

3.3 Сборка и запуск

Выполните последовательно:

```
docker compose --profile proxy build
docker compose --profile proxy up -d
docker compose ps -a
```

Сборка и запуск должны завершиться с кодом 0. Сервис prepare должен завершиться с кодом 0; db и web — перейти в состояние healthy; постоянные обработчики — оставаться запущенными. При ошибке prepare не открывайте установку пользователям. Получите ограниченный фрагмент журнала соответствующей службы:

```
docker compose logs --tail=100 prepare
docker compose logs --tail=100 web
```

Не запускайте демонстрационное заполнение базы в рабочем контуре. Не удаляйте существующий том БД для обхода ошибки прав или схемы. Если кластер уже содержит данные, первичная установка должна быть заменена согласованной процедурой обновления.

Проверьте изоляцию обработки документов в запущенной конфигурации:

```
docker compose exec -T web bin/rails documents:check_sandbox
docker compose exec -T web bin/rails documents:check_isolation
```

Ожидается успешное завершение обеих команд. При отказе проверьте применённые SELinux/seccomp-профили и диагностические сообщения. Не снимайте ограничения, чтобы получить успешный запуск обработки недоверенных файлов.

3.4 Первоначальный административный доступ

После проверки HTTPS создайте первое приглашение. Подставьте фактический адрес ответственного администратора вместо admin@example.org:

```
docker compose run --rm \
  -e BOOTSTRAP_EMAIL=admin@example.org \
  -e BOOTSTRAP_NAME=Администратор \
  web bin/rails staff:bootstrap
```

Команда выводит одноразовую ссылку со сроком 24 часа. Это секрет доступа: передайте её адресату лично. Администратор открывает ссылку, задаёт уникальный пароль не короче 16 символов, подключает TOTP по QR или ручному ключу и вводит текущий шестизначный код. Ожидается успешный вход в административный кабинет. После подключения первого администратора повторный bootstrap больше не создаёт учётные записи.

В разделе «Сотрудники» пригласите второго администратора и персонал. Назначьте роли и туры через «Доступ и туры». Координатор видит все туры; оператор, жюри и комиссия — только назначенные. Проверьте права отдельным входом каждой роли. Для утраты второго фактора используйте штатный сброс другим администратором; публичного обходного входа нет. Восстановление последнего администратора требует отдельного регламента владельца установки.

Координатор создаёт сезон и туры, задаёт сроки показа и приёма апелляций. Проверяйте время в UTC+5, включая исключительную конечную границу. Закрытие сезона прекращает доступ участников, но не удаляет архив материалов.

4 ПРОВЕРКА ПРОГРАММЫ

4.1 Приёмочная проверка установки

Используйте отдельный учебный сезон и вымышленные данные. Рабочие данные участников в контрольный пример не включайте. На внешнем рабочем месте выполните запрос к своему домену:

```
curl --fail --silent --show-error \
  https://vsoshlk.example.org/up
```

Ожидается HTTP 200 и действительный сертификат без отключения его проверки. При ошибке проверьте DNS, доступность 443, сроки сертификата, соответствие PUBLIC_HOST/PUBLIC_URL и состояние web. Ответ /up подтверждает только ограниченную проверку веб-приложения.

Далее проверьте полный путь данных: координатор создаёт учебный тур, импортирует учебного участника и результаты, загружает работу, критерии и разбор; обработчик завершает подготовку; участник входит и получает только свою работу. По прямой ссылке чужого участника доступ должен быть запрещён. Участник подаёт учебную апелляцию; координатор публикует решение и протокол; участник видит решение и подтверждает ознакомление. Запишите фактическую версию, время, исполнителя и результат каждого действия.

Отдельно проверьте почту на согласованном учебном адресе, наличие файлов в кабинете после перезапуска контейнеров и контрольное восстановление копии по разделу 5.2. До выполнения этих проверок установка не считается принятой только по состоянию healthy.

4.2 Текущий контроль очередей

Из корня установки на узле выполните:

```
timeout 30s docker compose exec -T \
  web bin/rails operations:check
```

Команда выдаёт JSON и не меняет задания. Поле status принимает ok, attention или unavailable; database отражает доступность БД; queues содержит counts, overdue_queued и long_running по видам задач. Код 0 означает отсутствие обнаруженных отклонений именно в проверяемых очередях; 1 — ошибочные задания, ожидание свыше 15 минут или обработку свыше 30 минут; 2 — ошибку обращения к БД. Код 124 относится к внешней команде timeout. Остальные коды, например ошибка Docker или отсутствие задачи в образе, разбираются отдельно и не считаются кодами проверки приложения.

Контролируйте HTTPS, свободное место, сигналы процессов, возраст очередей, почту и копии раздельно. Периодичность и пороги свободного места назначает ответственный по нагрузке площадки. Проверка operations:check сама не проверяет сигналы процессов, SMTP, файловый том, внешние способы входа и резервную копию.

В кабинете «Задачи» проверьте состояние, время начала и число попыток задания; сравните два наблюдения. Долгая обработка не доказывает зависание. При attention выясните, какой вид очереди вызвал отклонение, проверьте соответствующий сервис через docker compose ps и его журнал через docker compose logs --tail=100 ИМЯ_СЕРВИСА. Повторяйте задание штатной командой интерфейса после устранения причины и проверки отсутствия активного выполнения. Прямое изменение состояния очереди через SQL не является процедурой восстановления.

5 СОПРОВОЖДЕНИЕ И РЕЗЕРВИРОВАНИЕ

5.1 Остановка и возобновление

Согласуйте окно обслуживания, прекратите приём новых запросов и дождитесь завершения активных операций. Затем остановите прикладные службы:

```
docker compose --profile proxy stop \
  proxy web worker registry scheduling mail
```

Если отдельно включён max, остановите и его. Сервис db и сохраняемые тома остаются на месте. Перед возобновлением проверьте готовность БД, конфигурацию и необходимые секреты, затем выполните docker compose --profile proxy up -d и проверки раздела 4. Команды удаления томов для штатной остановки не применяются.

5.2 Создание и проверка резервной копии

Резервная копия содержит согласованный снимок PostgreSQL и всех объектов Active Storage Disk: оригиналы, сканы и производные файлы. Конфигурацию, точную поставку, необходимые credentials и версии SECRET_KEY_BASE сохраняют дополнительно в защищённых хранилищах. Секреты не включаются в открытый архив с документацией.

На узле создайте отдельный каталог для копий, доступный UID 1000 образа:

```
sudo install -d -m 0700 -o 1000 -g 1000 \
  /srv/vsoshlk-backups
```

Путь является примером; место должно иметь достаточный запас. Сервис backup базового compose.yml монтирует файлы приложения только для чтения, но не монтирует каталог /backups: его необходимо передать явно. Выполните:

```
docker compose --profile ops run --rm \
  -v /srv/vsoshlk-backups:/backups:Z backup
```

В новой конфигурации сервис получает отдельную роль vsosh_restore с правом CREATEDB для временной проверочной БД на том же сервере. Эта роль отличается от рабочей, не имеет административных атрибутов и не передаётся web. При SELinux используйте согласованную маркировку выделенного каталога; не применяйте :Z к системным или совместно используемым каталогам.

Ожидается код 0, новый подкаталог с уникальным именем и файлы manifest.json, database.dump.enc, каталог files с зашифрованными объектами, restore-check.json. В отчёте контрольного восстановления должны быть подтверждены проверка шифрованной копии, хешей/строк, сброса сессий и карантина доставки. Наличие одного дампа или каталога без manifest.json не является завершённой копией.

Штатная проверка создаёт собственную временную БД, восстанавливает туда дамп и файлы, сверяет данные, проверяет подготовку после восстановления и удаляет только созданную ею БД. Исходная рабочая БД не переключается. Для повторной проверки готового каталога используйте ту же поставку и исходный секрет:

```
docker compose --profile ops run --rm \
  -v /srv/vsoshlk-backups:/backups:Z \
  backup bin/restore_check /backups/ИМЯ_КОПИИ
```

Подставьте точное имя завершённого каталога. Ожидается успешный JSON-отчёт и код 0. При ненулевом коде копию не отмечайте как проверенную; сохраните прежние успешные копии, разберите сообщение по разделу 7. Копирование не удаляет предыдущие копии автоматически. Вынос на другой

узел, расписание, срок хранения и оповещение о сбое настраиваются отдельно. Для каждого зашифрованного файла и дампа действует предел 32 ГБ; резервирование другого типа хранилища в этой поставке не поддерживается.

5.3 Аварийное восстановление

Проверка bin/restore_check не выполняет аварийное переключение. Возврат данных на рабочий или новый узел требует плана конкретной площадки: выбранной копии, совместимой версии программы/PostgreSQL, отдельной целевой БД, отдельного файлового тома, исходных ключей, сетевой изоляции и ответственного за открытие доступа. Непроверенный перенос на другой узел не имеет гарантированных RTO/RPO.

До начала работ зафиксируйте имя и адрес исходной БД и отдельно целевой восстановленной БД. Остановите все прикладные подключения к цели и исключите новые подключения правилами сети. Сначала проверьте копию в изолированном контуре по разделу 5.2. Импорт зашифрованного комплекта должен возвращать совместно дамп и весь набор файлов с проверкой их хешей. Простая передача database.dump.enc в pg_restore невозможна: это зашифрованный файл, требующий исходного SECRET_KEY_BASE и штатного формата расшифрования BackupBundle.

После возврата данных и файлов, до старта web и обработчиков, выполните подготовку восстановленной базы. Ниже — команда для уже подготовленного изолированного контейнера совместимой поставки. DATABASE_URL и PGPASSWORD задаются ему отдельно и указывают на цель; их значения не берутся из рабочей установки автоматически:

```
RESTORED_DATABASE_NAME=vsosh_recovered \
  bin/rails recovery:prepare
```

Замените vsosh_recovered точным именем целевой БД. Команда откажет при несовпадении имени или наличии других клиентских подключений. Текущая схема должна содержать миграцию 20261024011000; старую копию сначала проверяют совместимым выпуском и затем проводят проверенный путь миграций.

Ожидается JSON-отчёт: прежние LoginSession/OauthAttempt удалены; недоставленная почта остановлена с причиной recovery; незавершённые уведомления MAX отменены, прежние активные подключения/ссылки MAX отключены; события записаны в аудит. Команда не переключает прокси, не импортирует дамп, не восстанавливает файлы и не отзывает постоянные пароли, PIN, внешние связи входа, приглашения и ключи сканера.

До открытия трафика ответственный сверяет постоянные доступы, отзывает восстановленные устаревшие полномочия, проверяет файлы и роли, проходит контрольный сценарий и только затем разрешает подключения. Старые письма не выпускают из карантина автоматически: требуется сверка с внешней системой доставки. Повторно выполнять recovery:prepare после открытия нельзя — будут сброшены и новые сессии/согласия. Полный импорт и переключение должны быть предварительно отрепетированы на конфигурации площадки; настоящий раздел не является протоколом такого испытания.

5.4 Обновление

До обновления сохраните успешную проверенную копию, конфигурацию и идентификаторы прежних образов. На изолированной копии проверьте миграции и контрольный сценарий. Согласуйте окно, остановите прикладные службы по 5.1, установите доверенную новую поставку и соберите образы. Выполните

подготовку схемы новым сервисом prepare, проверьте код 0 и затем запустите web и все обработчики нового выпуска. Проверьте раздел 4 и сохраните журнал результата.

Возврат только старого образа допускается при подтверждённой совместимости схемы. Если схема изменилась несовместимо, необходим согласованный возврат БД и файлов с учётом изменений, возникших после копии. Не применяйте down -v и не выполняйте произвольный db:rollback как универсальную процедуру отката.

6 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Администратор приложения выбирает профиль модулей в интерфейсе. Отключение панели задач не останавливает обработчик документов. Проверяйте выбранный профиль на учебном сезоне и назначайте права до работы персонала.

Внешние способы входа Яндекс ID, VK ID, Сбер ID и ЕСИА требуют регистрации приложений, согласованных callback-адресов и защищённых реквизитов. MAX требует собственной регистрации и явного запуска профилей max/max-setup. Их параметры и внешний приёмочный сценарий зависят от зарегистрированного приложения; включать их только после отдельной настройки и проверки. Они не требуются для автономного сценария публикации и апелляций.

Рабочее место сканера подключается через выдаваемый программой комплект агента на конкретный тур. Для каждого места создаётся собственный ключ; утраченный комплект отзывают. Агент и исходные сканы размещают в закрытой папке оператора, конфигурацию и квитанции — отдельно от входящей папки сканера. Порядок пользовательского подключения приведён в руководстве оператора.

7 СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

«status: attention» (JSON operations:check), код 1: есть failed-задания, слишком старое ожидание или длительная обработка. Проверьте вид очереди, состояние соответствующего сервиса и прогресс. После устранения причины выполните разрешённый повтор и повторите контроль. Порог длительности не означает, что процесс следует немедленно завершить.

«status: unavailable», код 2: запрос к БД не выполнен. Проверьте db, доступность сети контейнеров, рабочую роль и состояние диска; затем повторите проверку. Поле error_class сообщает класс исключения; текст ошибки может содержать чувствительные сведения и не публикуется.

Код 124 внешнего timeout: пробник не завершился за 30 секунд. Проверьте состояние Docker, загрузку узла и доступность БД. Не подменяйте этот код успешным и не увеличивайте таймаут без выяснения причины.

«Нет полного манифеста» / «Копия неполна или повреждена»: копирование не завершено либо отсутствуют/повреждены файлы. Сравните с журналом создания; выберите другую завершённую копию. Не создавайте manifest.json вручную.

«Подпись резервной копии не совпала»: неверный исходный SECRET_KEY_BASE либо изменённый манифест. Проверьте происхождение и версию секрета. Не отключайте проверку подписи и не пересоздавайте её для неизвестной копии.

«Контрольная сумма не совпала» / «Расшифрованный файл не совпал»: содержимое копии повреждено либо подменено. Не продолжайте импорт; используйте другой проверенный экземпляр и исследуйте хранилище.

«Ошибка расшифрования копии»: неверные ключи или повреждённые параметры/данные шифрования. Проверьте исходный секрет и целостность копии. Неверный ключ нельзя заменить новым случайным значением.

«Нужны pg_dump и pg_restore той же основной версии, что PostgreSQL»: версия клиента не совпадает с сервером. Используйте совместимую поставку PostgreSQL 16; не меняйте манифест для обхода контроля.

«Поддерживается только Disk-хранилище»: текущая реализация резервирования не поддерживает выбранный сервис файлов. Не считайте один дамп достаточной копией; сначала обеспечьте проверенный способ сохранения всех файлов.

«Укажите точное имя восстановленной БД»: не задано или неверно RESTORED_DATABASE_NAME. Сверьте адрес и имя цели; исправьте окружение только после подтверждения, что это изолированная восстановленная БД.

«Остановите все подключения к восстановленной БД»: обнаружены другие клиенты. Остановите прикладные процессы и закройте новые подключения; затем повторите подготовку. Сама команда не завершает чужие подключения.

При неописанном сообщении сохраните время, идентификатор выпуска, имя службы, код завершения и обезличенный фрагмент журнала. Обращение в поддержку: admin@edzsoft.ru. Пароли, ключи, .env, СНИЛС и изображения работ в открытое обращение не включать.