

Руководство администратора

Платформа для управления железнодорожными грузоперевозками «Рейлог (Railog)»

На 19 листах

ООО «Рейлог» – один из ведущих разработчиков программного обеспечения для автоматизации железнодорожной логистики. Одним из продуктов компании является платформа «Рейлог (Railog)» – российское программное решение, предназначенное для комплексного управления логистическими процессами на железнодорожном транспорте.

ООО «Рейлог»

Офис: 142000, Московская обл., г. Домодедово, Центральный мкр., ул. Станционная, д. 3, ПОМЕЩ. 207.

Эл. почта: info@railog.ru

Телефон: +7 (499) 113 13 26

ОГЛАВЛЕНИЕ

Введение	4
О платформе «Рейлог»	4
Уровень подготовки персонала	5
Выполняемые работы	5
Предварительные требования	6
Установка и обновление	6
Установка и запуск на Linux с использованием Wine	8
Проверка работоспособности для Linux (при использовании PostgreSQL)	14
Предварительные требования	14
Основные команды Docker Compose	14
Запуск и остановка сервисов	14
Просмотр состояния и логов	15
Команды для отдельных контейнеров	16
Команды для обновления контейнеров	18
Диагностика и устранение проблем	18
Дополнительные команды	18
Проверка работоспособности для Windows (при использовании SQL Server)	19
Обращение в Службу технической поддержки	19

Введение

О платформе «Рейлог»

Платформа для управления железнодорожными грузоперевозками «Рейлог (Railog)» – российское программное решение, предназначенное для комплексного управления логистическими процессами на железнодорожном транспорте.

Платформа включает как общие, так и специализированные решения для различных участников отрасли, работающих с железнодорожным парком:

- компании, эксплуатирующие подвижной состав (операторы);
- собственников/арендаторов вагонов;
- экспедиторские и терминально-складские компании (владельцы п/путей, ГРО и ГРП).

Ключевые возможности платформы «Рейлог»:

- Консолидация данных из различных источников (включая отраслевые системы) и предоставление структурированную информацию для анализа и управления;
- Автоматизация процессов: формирование электронных документов, контроль сроков, управление ремонтами вагонов, учет арендованного парка, расчет ставок;
- Сквозное управление перевозками – от планирования маршрутов до контроля выполнения обязательств;
- Интеллектуальная обработка данных и событийно-ориентированная автоматизация процессов;
- Анализ данных и прогнозирование;
- Сбор и хранения исторических и аналитических данных перевозочного процесса.

Платформа «Рейлог» решает следующие задачи:

- Автоматизация производственных процессов железнодорожных грузоперевозок;
- Оперативный анализ производственных процессов;
- Интеграция с автоматизированной системой подготовки и оформления перевозочных документов на железнодорожные грузоперевозки ОАО «РЖД» по территории Российской Федерации (ЭТРАН);
- Повышение операционной эффективности предприятия;
- Снижение количества ошибок при подготовке отчетов и оформлении документов;
- Минимизация ручного труда в управлении перевозками.

Уровень подготовки персонала

Персонал платформы «Рейлог» имеет следующие категории:

- персонал, осуществляющий непосредственную эксплуатацию платформы «Рейлог» (пользователи);
- обслуживающий персонал платформы «Рейлог», осуществляющий техническое обслуживание (администратор).

Администратор должен обладать следующими знаниями и навыками:

- свободно ориентироваться в программно-технической документации;
- свободно ориентироваться в стандартных возможностях используемых ОС и СУБД (реляционных и не реляционных), протоколах передачи данных;
- владеть средствами мониторинга СУБД;
- владеть средствами резервного копирования и восстановления данных;
- владеть средствами защиты информации;
- работать с архиваторами, дисковыми утилитами, антивирусными программами;
- определять источник сбоя функционирования ПО и грамотно описывать его;
- иметь навыки работы с серверным и телекоммуникационным оборудованием;
- иметь расширенные знания в области поддержки пользователей.

Перед вводом платформы «Рейлог» в промышленную эксплуатацию администратор должен изучить эксплуатационную документацию.

Для работы с платформой «Рейлог» администратору необходимо ознакомиться с настоящим Руководством администратора.

Выполняемые работы

Администратор выполняет следующий перечень работ для настройки, ввода в эксплуатацию и поддержания работоспособности платформы «Рейлог»:

- установка компонентов служебного ПО «Рейлог» (подробное описание приведено в Руководстве по развертке «Рейлог»);
- подготовка базы данных (подробное описание приведено в Руководстве по развертке «Рейлог»);
- смена каталога для БД PostgreSQL (при необходимости) (подробное описание приведено в Руководстве по развертке «Рейлог»);
- обновление структур базы данных (подробное описание приведено в Руководстве по развертке «Рейлог»);
- установка служебного ПО «Рейлог» на Windows сервер (при необходимости) (подробное описание приведено в Руководстве по развертке «Рейлог»);

- интеграция с ЭТРАН (подробное описание приведено в Руководстве по развертке «Рейлог»);
- настройка автообновления (подробное описание приведено в Руководстве по развертке «Рейлог»);
- настройка почтового сервера (подробное описание приведено в Руководстве по развертке «Рейлог»);
- настройка бекапов (подробное описание приведено в Руководстве по развертке «Рейлог»);
- установка и обновление (см. п. Установка и обновление);
- установка и запуск на Linux с использованием Wine (см. п. Установка и запуск на Linux с использованием Wine);
- проверка работоспособности БД (см. п. Проверка работоспособности БД).

Предварительные требования

Для функционирования ПО рабочих мест платформы «Рейлог» предъявляются следующие требования к техническому обеспечению:

- 64-битная архитектура (допускается использование 32-битной архитектуры);
- не менее 8Гб оперативной памяти;
- SSD не менее 200Гб;
- ОС Windows 10 (1809 и выше);
- Допустимо использование более ранних версий, начиная с Windows 7 (неофициальная поддержка). Также поддерживаются рабочие места на Linux с использованием Wine.

Установка и обновление

Перед установкой ПО рабочих мест платформы «Рейлог» необходимо установить .net 8.0 (или более поздних версий 8.*). Для функционирования необходимо установить следующие компоненты:

- .NET Desktop Runtime;
- ASP.NET Core Runtime.

Полный список .net core компонентов доступен по ссылке: <https://dotnet.microsoft.com/download/dotnet/8.0>.

Для установки необходимо выбрать и установить версию соответствующей архитектуры (x64 или x86) (см. Рисунок 1).



Рисунок 1 – Установка компонентов .net core 8.0

После установки компонентов .net core 8.0 необходимо запустить установочный пакет последней версии ПО рабочих мест платформы «Рейлог», имеющий вид «Railog-***-x643.*.*».

Критерии определения установочного пакета:

- пакет с расширением .msix устанавливается на Windows 10.
- пакет с расширением .msi устанавливается на более ранних версиях Windows, а также на серверных версиях Windows.
- постфикс -x64 и -x86 используется для выбора разрядности, соответствующей архитектуры системы 64 и 32-битные соответственно.

После завершения установки можно запустить исполняемый файл Railog.exe ПО рабочих мест платформы «Рейлог» из меню «Пуск». В случае наличия проблем запуска необходимо следовать предлагаемым инструкциям по установке пропущенных компонентов.

В случае успешного запуска появится стартовое окно ПО рабочих мест платформы «Рейлог» (см. Рисунок 2).

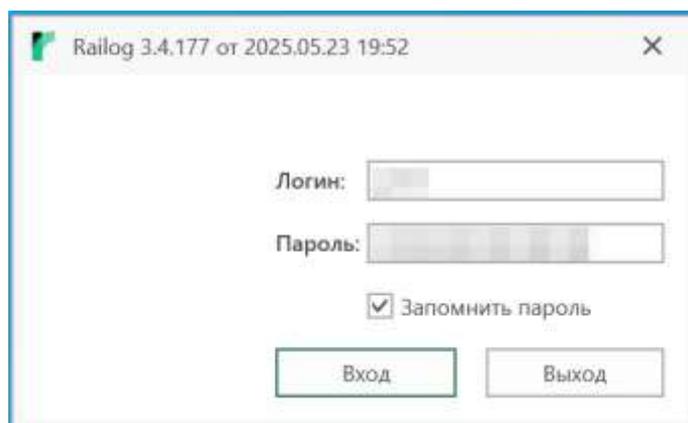


Рисунок 2 – Стартовое окно

Для обновления программы необходимо запустить установочный пакет нужной версии ПО рабочих мест платформы «Рейлог».

В случае использования ПО автообновления, при перезапуске ПО рабочих мест платформа «Рейлог» выполняет автоматическую проверку и установку обновлений, ручной запуск установочного пакета при этом не требуется.

Установка и запуск на Linux с использованием Wine

Шаг 1. Установка Linux.

В процессе установки выбрать уровень защищенности Базовый «Орел».

Выбрать региональные настройки по умолчанию.

Установить компоненты Linux, указанные на рисунке ниже (см. Рисунок 3).

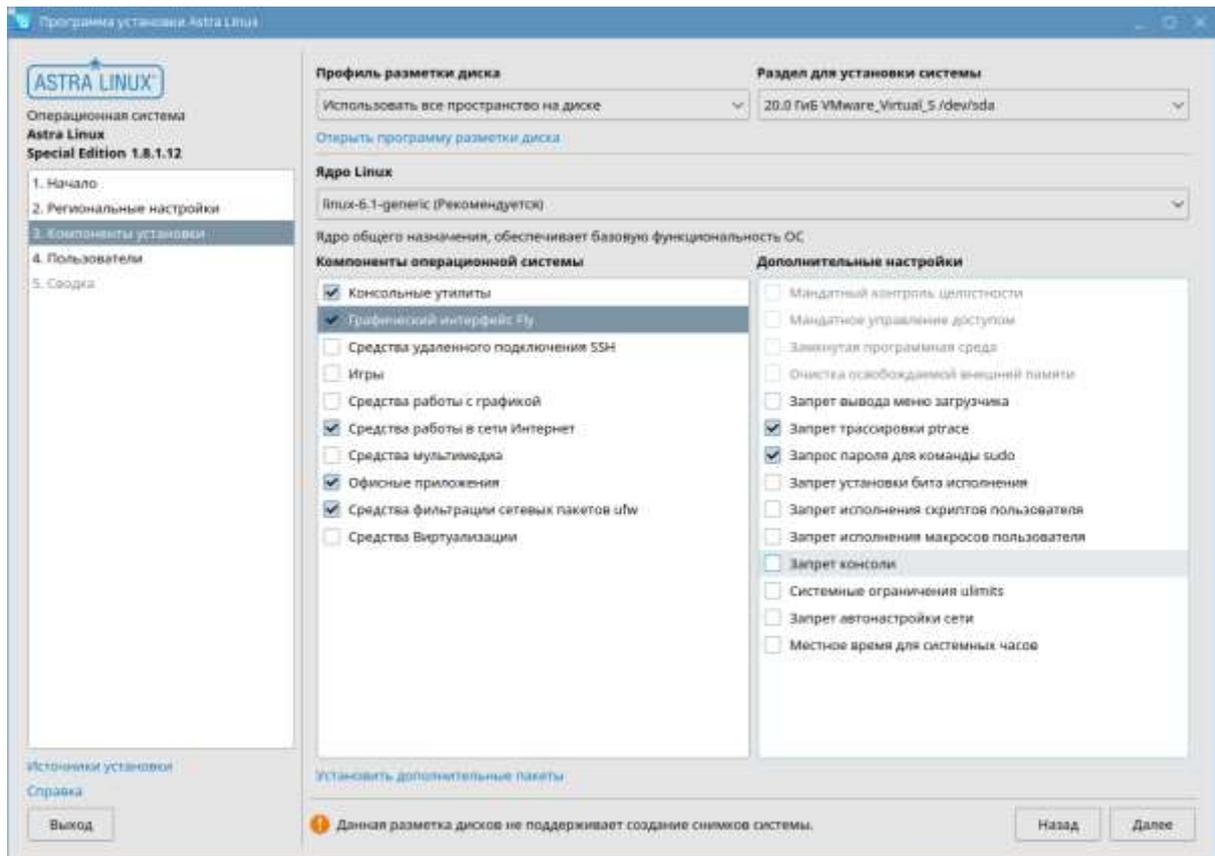


Рисунок 3 – Компоненты Linux

Далее использовать пользователя administrator (root).

Шаг 2. Настройка Wine

Для настройки Wine необходимо выполнить следующие действия:

1. Запустить менеджер пакетов Synaptic.
2. Выбрать «Настройки» -> «Разрешаем все репозитории».
3. Использовать команду `sudo apt install wine`.
4. Использовать команду `sudo apt install winetricks`.
5. Использовать команду `sudo winetricks --self-update`.
6. Использовать команду `sudo apt install zenity`.
7. Запустить winetricks, который автоматически просит установить Wine-Mono (см. Рисунок 4).

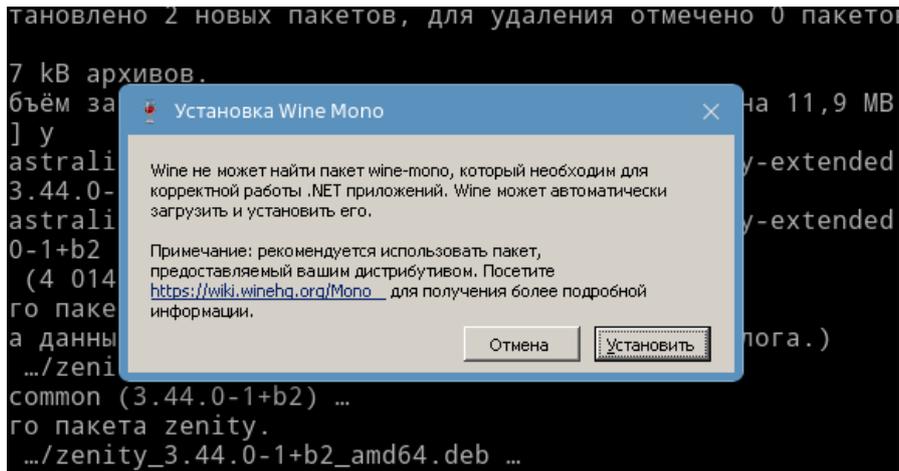


Рисунок 4 – Установка Wine Mono

8. Нажать «Установить».
9. Далее для проверки работы Wine выбрать «Использовать префикс по умолчанию» -> «ОК» -> «Запустить winescfg». Открывается окно настройки Wine (см. Рисунок 5).

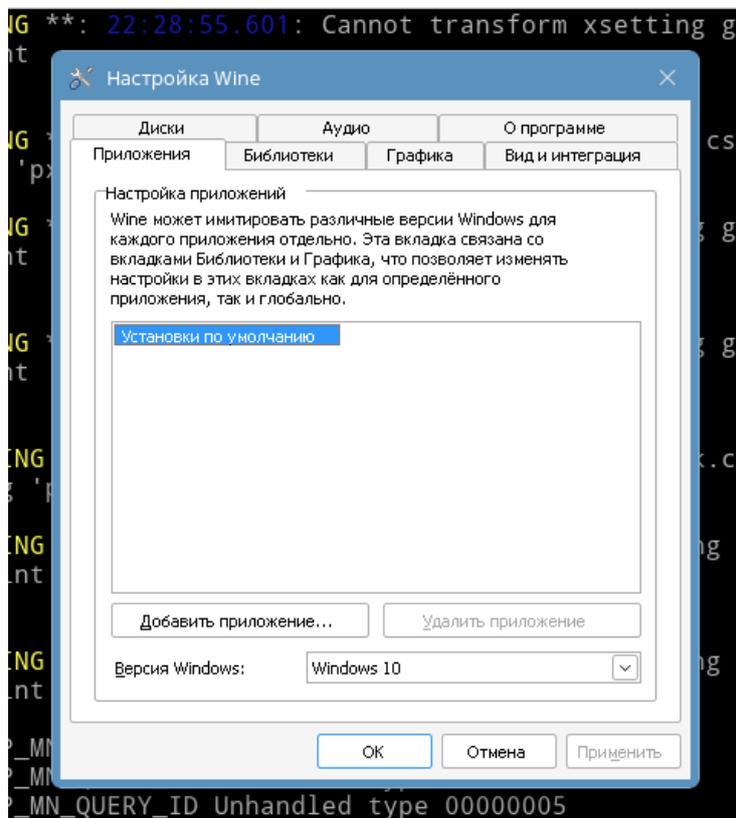


Рисунок 5 – Окно настройки Wine

10. Для закрытия winetricks нажимать везде «Отмена».
- Шаг 3. Копирование «Рейлог» на Linux-хост.

Если установка происходит в VM VMWare, то процесс производить через менеджер пакетов Synaptic open-vm-tools open-vm-tools-desktop следующим образом:

1. В VMWare Workstation создать общую папку для VM и задать ей наименование, например «vm».
2. Для проверки доступна ли папка в Linux выполнить команду `vmware-hgfsclient`, наименование папки должно быть в выводе команды.
3. Если не существует, то создать `sudo mkdir /mnt/hgfs`.
4. Выполнить монтирование `sudo vmhgfs-fuse .host: /mnt/hgfs -o allow_other -o uid=1000`.
5. Проверить доступно ли содержимое папки.
6. Чтобы эта папка монтировалась всегда при запуске добавляем в `/etc/fstab` строку `.host: /mnt/hgfs fuse.vmhgfs-fuse defaults,allow_other,uid=1000,nonempty 0 0`.

Если установка происходит вне VM VMWare, то необходимо копировать папку «Railog» в `/home/папка пользователя/.wine/drive_c`.

Шаг 4. Запуск исполняемого файла

Для запуска исполняемого файла `Railog.exe` необходимо выполнить следующие действия:

1. Запустить через консоль `wine start 'c:\Railog\Railog.exe'` при первом запуске просит установить .Net Desktop Runtime x64 8.0.8 (см. Рисунок 6).

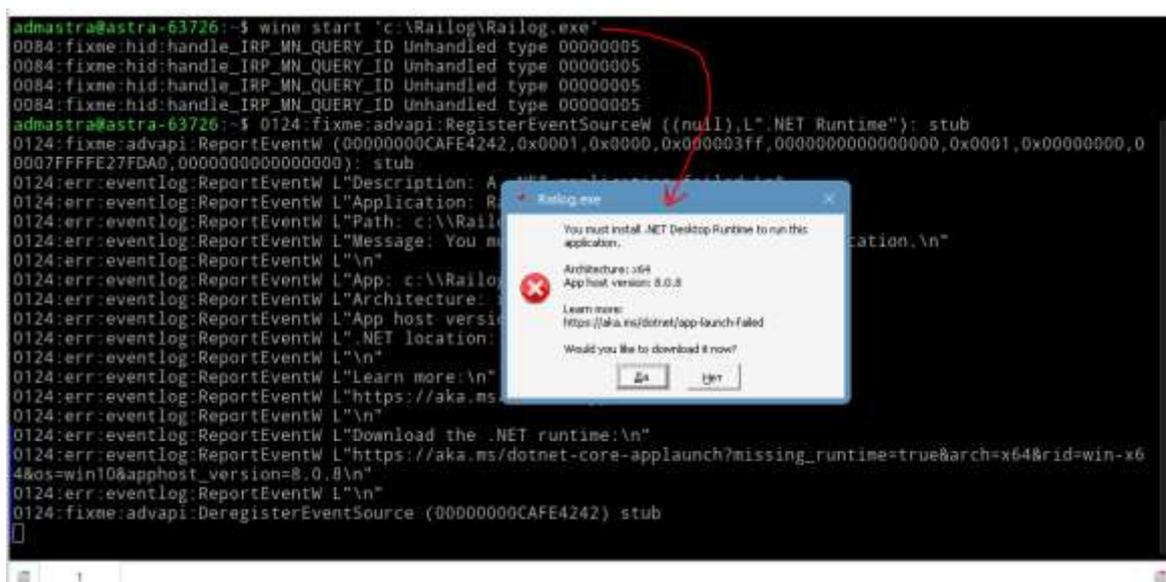


Рисунок 6 – Запуск через консоль `wine start «c:\Railog\Railog.exe»`

2. Загрузить установщик, переименовать, например, в `net64808.exe` и перенести в `/home/папка пользователя/.wine/drive_c`.
3. Установить .Net Desktop Runtime командой `wine start 'c:\Railog\net64808.exe'`.

4. Запустить через консоль `wine start 'c:\Railog\Railog.exe'`, при втором запуске просит установить Microsoft.AspNetCore.App 8.0.0 (см.).

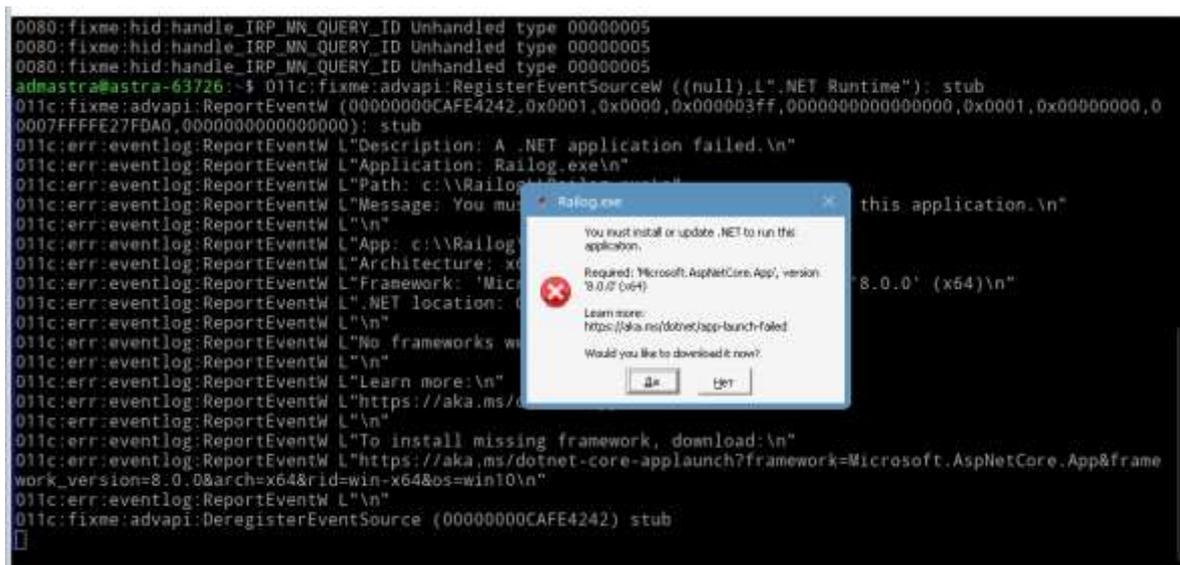


Рисунок 7 – Запуск через консоль `wine start 'c:\Railog\Railog.exe'`

5. Загрузить установщик, переименовать, например, в `asp64800.exe` и перенести в `/home/папка пользователя/.wine/drive_c`.
6. Установить `Microsoft.AspNetCore.App` командой `wine start 'c:\Railog\asp64800.exe'`.
7. Запустить через консоль `wine start 'c:\Railog\Railog.exe'` На этом этапе запускается окно ввода пароля (см. Рисунок 8).
8. После ввода пароля выполняется загрузка справочников, открывается основное окно и сразу закрывается, в консоли ошибки типа `Failed to get device for window`.
9. Запустить `winetricks` → «Использовать префикс по умолчанию» → «Установить шрифт». Установить галку «allfonts» и нажимаем «ОК». Далее происходит долгая установка, нужно много раз нажимать кнопку «ОК» (примерно 10 минут). Для оптимизации установки необходимо уточнить какие шрифты необходимы для работы ПО «Рейлог».
10. Запустить через консоль `wine start 'c:\Railog\Railog.exe'`, ввести пароль, ожидать загрузки данных. Открывается основное окно, подгружается меню. «Рейлог» запущен, можно проверять работу реестров (см. Рисунок 9).

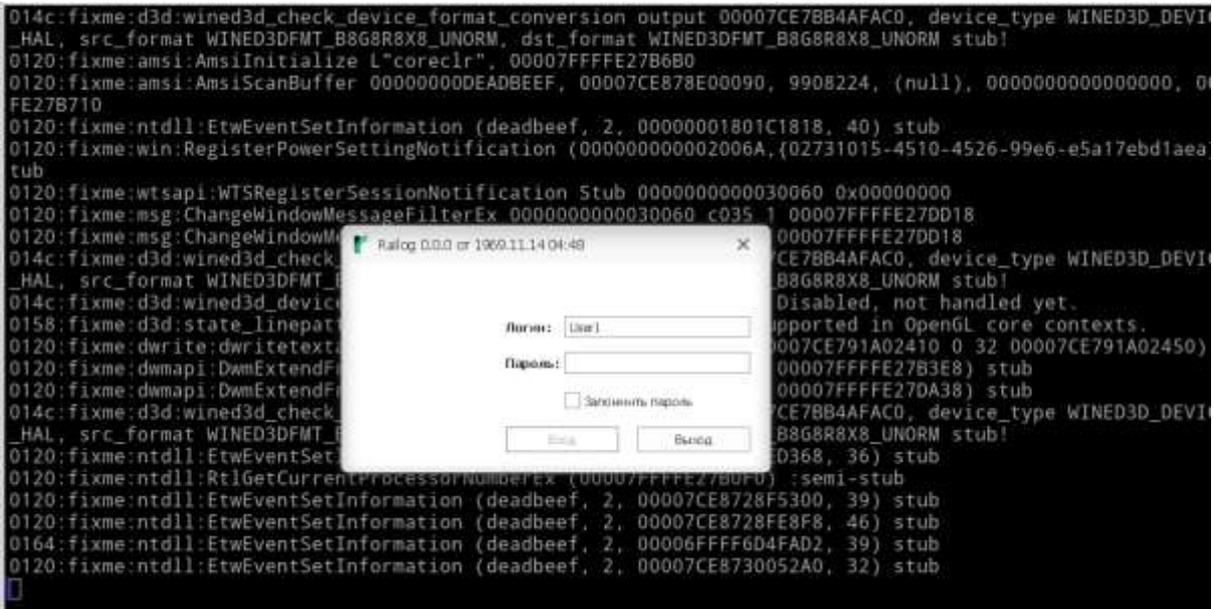


Рисунок 8 – Окно ввода пароля

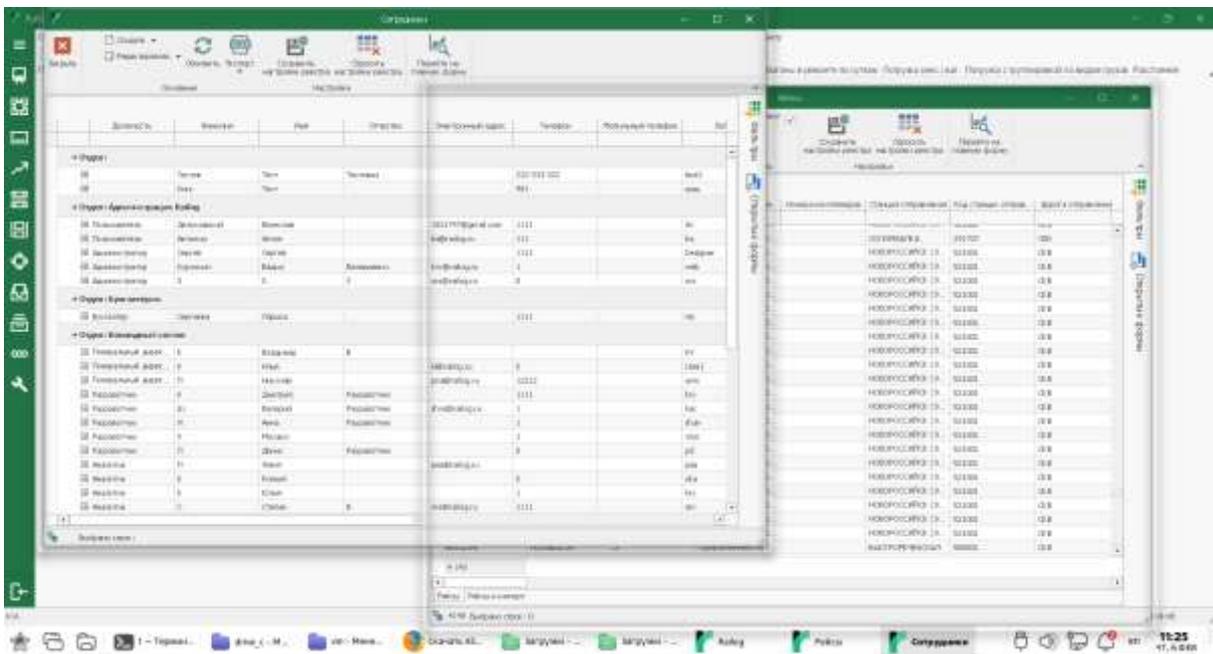


Рисунок 9 – «Рейлог» запущен

Если запускается с ошибкой «Major opcode of failed request: 152 (XFree86-VidModeExtension)», то задать следующие переменные окружения:

[HKEY_CURRENT_USER\Software\Wine\X11 Driver]

"UseXRandR"="N"

"UseXVidMode"="N"

Проверка работоспособности для Linux (при использовании PostgreSQL)

Предварительные требования

Проверка работоспособности осуществляется в Docker и Docker Compose.

Файл `docker-compose.yml` должен находиться в рабочей директории.

При отсутствии прав может потребоваться выполнять команды от имени суперпользователя, тогда перед командой необходимо писать `sudo`, например: `sudo cat docker-compose.yml`.

Файл конфигурации должен находиться в `/docker-data`. Каталог может быть выбран любой.

Перед выполнением команд `docker-compose` необходимо перейти в этот каталог: `cd /docker-data`.

Для просмотра содержимого `docker-compose.yml` можно использовать команду: `cat docker-compose.yml`.

Основные команды Docker Compose

Запуск и остановка сервисов

1. Запуск всех сервисов (в фоновом режиме): `docker-compose up -d`.

Параметры: `-d` (`--detach`) – запуск в фоне (без блокировки терминала).

Результат:

- Создаются и запускаются все контейнеры из `docker-compose.yml`.
- Вывод содержит список контейнеров и их статус (`Created / Started`).

Пример ответа:

```
Creating postgres17 ... done
```

```
Creating test_import ... done
```

```
Creating test_etran ... done
```

2. Остановка всех сервисов: `docker-compose down`.

Параметры:

- Без параметров – останавливает и удаляет контейнеры, сети и тома (если не указано иное).

- -v – дополнительно удаляет тома (volumes).

Результат:

- Вывод показывает остановку и удаление каждого контейнера:
Stopping test_import ... done
Removing test_import ... done
Removing network default

3. Перезапуск всех сервисов: `docker-compose restart`

Параметры:

- Без параметров – перезапускает все контейнеры.
- Можно указать конкретный сервис: `docker-compose restart test_import`.

Результат: для каждого контейнера выводится `Restarting <имя> ... done`.

4. Остановка/запуск конкретного сервиса:

- `docker-compose stop test_import`
- `docker-compose start test_import`

Результат: возвращает имя контейнера (`test_import`), если команда выполнена успешно.

Просмотр состояния и логов

1. Просмотр статуса всех контейнеров: `docker-compose ps`.

Результат: таблица с состоянием контейнеров (см. Таблица 1).

Таблица 1 – Состояние контейнеров

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS	PORTS
postgres17	postgres:17	docker-entrypoint.s	postgres	6 weeks ago	Up 10 days	0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp
test_import	import:latest	dotnet Import.dll	test_import	9 days ago	Up 9 days	80/tcp, 443/tcp

Описание полей:

- NAME – название контейнера.
- IMAGE – используемый образ.
- COMMAND – команда запуска контейнера.
- SERVICE – название сервиса в `docker-compose.yml`.

- **CREATED** – время создания контейнера.
- **STATUS** – текущий статус (Up — работает, Exited — остановлен).
- **PORTS** – проброшенные порты (формат: хост:контейнер).

2. Просмотр логов сервиса: `docker-compose logs -f test_import`

Параметры:

- `-f (--follow)` – потоковый вывод (обновляется в реальном времени).
- `--tail=100` – показать последние 100 строк.

Результат: выводятся логи приложения с временными метками.

Для остановки просмотра логов сервиса нажать `Ctrl+C`.

Команды для отдельных контейнеров

1. Просмотр всех контейнеров (включая остановленные): `docker ps -a`

Параметры: `-a (--all)` – показывает все контейнеры.

Результат: таблица с состоянием контейнеров (см. Таблица 2).

Таблица 2 – Состояние контейнеров

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
85dac35b29ea	updater:latest	dotnet Updater.dll	9 days ago	Up 9 days	80/tcp, 443/tcp	test_updater
2aade8baa2ad	postgres:17	docker-entrypoint.s	6 weeks ago	Up 10 days	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	postgres17

Описание полей:

- **CONTAINER ID** – уникальный идентификатор контейнера (сокращённый хэш).
- **IMAGE** – используемый образ.
- **COMMAND** – команда запуска контейнера.
- **CREATED** – время создания контейнера.
- **STATUS** – текущий статус (Up — работает, Exited — остановлен).
- **PORTS** – проброшенные порты (формат: хост:контейнер).
- **NAMES** – название контейнера.

2. Запуск/остановка контейнера:

- `docker start test_import`.

- `docker stop test_import.`
- `docker restart test_import.`

Результат: возвращает имя контейнера (`test_import`).

3. Просмотр логов контейнера: `docker logs -f test_import.`

Результат: аналогично `docker-compose logs`, но работает для любого контейнера (даже не из `docker-compose`), параметры идентичны.

4. Вход в контейнер (интерактивная сессия): `docker exec -it test_import /bin/bash.`

Параметры:

- `-it` – интерактивный режим с TTY.
- `/bin/bash` (или `/bin/sh`) – оболочка для входа.

Результат: открывает терминал внутри контейнера.

5. Проверка использования ресурсов: `docker stats`

Результат: динамически обновляемая таблица (см. Таблица 3).

Таблица 3 – Динамически обновляемая таблица

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
c96364ffe1a7	test_import	0.06%	376.1MiB / 3.823GiB	9.61%	864MB / 61.8MB	16.4MB / 15.9MB	33
2aade8baa2ad	postgres17	0.02%	111.5MiB / 3.823GiB	2.85%	362MB / 5.22GB	98.4MB / 1.26GB	6

Описание полей:

- CONTAINER ID – уникальный идентификатор контейнера (сокращённый хэш).
- NAME – название контейнера.
- CPU % – процент использования CPU (от общего доступного).
- MEM USAGE / LIMIT:
 - MEM USAGE – текущее потребление памяти.
 - LIMIT – лимит памяти, доступный контейнеру.
- MEM % – процент использования памяти от лимита.
- NET I/O – сетевой трафик (полученные/отправленные).
- BLOCK I/O – дисковые операции (чтение/запись)
- PIDS – количество процессов (или потоков) внутри контейнера.

Команды для обновления контейнеров

1. Обновление из Container Registry:

- `docker-compose pull somehub.ru/hutest_import`, где «somehub.ru/» – это адрес реестра контейнеров, где располагается образ (сообщается отдельно, при его наличии).
- `docker-compose up -d test_import`.

Результат: выводит процесс загрузки нового образа (Pulling, Downloaded).

2. Обновление из локального архива (.tar):

- `docker load -i import_latest.tar`.
- `docker-compose up -d test_import`.

Результат: после загрузки выводит «Loaded image: import:latest».

Диагностика и устранение проблем

1. Проверка статуса контейнеров: `docker-compose ps`.

Убедитесь, что все контейнеры в состоянии «Up».

2. Проверка логов на ошибки: `sudo docker-compose logs --tail=50 2>&1 | grep -i "error"`

Убедитесь, что записи отсутствуют.

3. Если контейнер не запускается:

- Проверить логи: `docker-compose logs test_import`

4. Если контейнер в цикле перезапуска:

- Временно отключить авто-перезапуск: `docker update --restart=no test_import`.
- Запустить вручную и проверить ошибки:
 - `docker start test_import`.
 - `docker logs -f test_import`

Дополнительные команды

1. Очистка неиспользуемых ресурсов: `docker system prune -a`.

2. Просмотр информации о контейнере: `docker inspect test_import`.

3. Просмотр используемых томов: `docker volume ls`.

4. Проверка сети: `docker network inspect <network_name>`.

Проверка работоспособности для Windows (при использовании SQL Server)

Рекомендуется периодически следить за наиболее нагруженными запросами.

В случае обнаружения проблем снижения производительности рекомендуется также проводить анализ активных выполняемых запросов к БД в момент снижения производительности.

Вспомогательные средства для SQL Server: <https://www.brentozar.com/first-aid/>.

Обращение в Службу технической поддержки

Если что-то пошло не так – свяжитесь со службой технической поддержки, отправив запрос специалистам на электронную почту help@railog.ru или по телефону +7(499)113-35-24. Время работы службы технической поддержки с 09.00 до 20.00 по московскому времени в будние дни.