

ТЕХНИЧЕСКИЙ РАЗБОР

# PostgreSQL для 1С: репликация и автоматический failover

Отказоустойчивая пара Postgres Pro 1С: синхронная репликация, VIP и переключение за минуту

---



Июль 2026

**itfresh.ru** · ИТ-аутсорсинг для юридических лиц

# Суть проблемы

Сервер СУБД под 1С в офисе на 30–50 рабочих мест — почти всегда единственная точка отказа: при выходе железа из строя учёт встаёт на часы, а восстановление из ночного бэкапа теряет день документов. Мы закрываем это парой серверов PostgreSQL с синхронной репликацией и автопереводом виртуального IP: простой — до минуты, потерь транзакций — ноль.

## Почему это важно бизнесу

- Час простоя базы 1С — остановка выписки счетов, отгрузок и кассы: прямые потери растут с каждым часом
- Восстановление из ночного бэкапа теряет день работы бухгалтерии — документы придётся вносить заново
- Лицензии сборок PostgreSQL для 1С — 0 ₽: вторая нода не требует затрат на СУБД, в отличие от MS SQL
- Российская СУБД из реестра ПО снимает санкционные риски и закрывает требования импортозамещения
- Плановые работы на сервере перестают требовать вечеров и выходных — переключение незаметно для людей



# Ключевые параметры реализации

## 17.9-3.1C

актуальная сборка PostgreSQL 17 с патчами 1C из списка поддерживаемых СУБД — берём её или Postgres Pro 1C по докам v8.1c.ru

## < 60 с

целевое время failover: keepalived переводит VIP, скрипт промоутит реплику без участия админа наш стандарт

## 2 с × 3

интервал проверки pg\_isready и порог fall в vrrp\_script — VIP уезжает после 3 провалов подряд keepalived, наш стандарт

## 8-16 ГБ

wal\_keep\_size на primary — запас WAL под закрытие месяца, дополнительно к слоту репликации по докам PostgreSQL 17

## 25% RAM

shared\_buffers; effective\_cache\_size — не менее половины RAM по рекомендации Postgres Pro для 1C по докам postgrespro.ru

## 0 ₽

лицензии сборок PostgreSQL для 1C: и primary, и синхронный standby разворачиваем бесплатно 1c.postgres.ru



# Синхронная потоковая репликация primary → standby

## Что настраиваем

Пара серверов СУБД pg-1c-01/02, выделенный линк 10GbE только под трафик репликации

## Как мы это делаем

- 1 На primary: `wal_level=replica`, `max_wal_senders=10`, `wal_keep_size=8GB`, `synchronous_commit=on` и `synchronous_standby_names='pg1c02'` — имя совпадает с `application_name` реплики
- 2 В `pg_hba.conf` открываем `host replication` только с IP второй ноды, аутентификация `scram-sha-256` под отдельным пользователем репликации
- 3 Реплику снимаем `pg_basebackup -R -X stream -C --slot=pg02`: утилита создаёт `standby.signal` и пишет `primary_conninfo` (в нём задаём `application_name=pg1c02`) в `postgresql.auto.conf`
- 4 Физический слот гарантирует: primary не удалит WAL-сегменты, пока standby их не забрал, — реплика переживает свои перезагрузки
- 5 Приёмка: в `pg_stat_replication` `state=streaming`, `sync_state=sync`, лаги `write/flush/replay` в пределах миллисекунд

## РЕЗУЛЬТАТ

RPO = 0: подтверждённые документы 1С не теряются при отказе primary. Реплика в `hot_standby` принимает `read-only` подключения — на ней же снимаем бэкапы, не нагружая боевой сервер.

## КЛЮЧЕВОЙ НЮАНС

`synchronous_standby_names` по умолчанию пуст — без него `synchronous_commit=on` подтверждает коммит только локально, и «синхронная» реплика на деле асинхронная. Проверяем `sync_state`, а не верим конфигу.

# Автоматический failover: keepalived + скрипт промоушена

## Что настраиваем

Обе ноды СУБД + виртуальный IP, к которому подключается кластер серверов 1С:Предприятие 8.3

## Как мы это делаем

- 1 `vrrp_script`: проверка `pg_isready` каждые 2 с, `fall=3`, `rise=2`; `track_script` снижает `priority` мастера после трёх провалов подряд — VIP уезжает на `standby`
- 2 На `standby` отработывает `notify_master`: скрипт вызывает `pg_ctl promote`, дожидается выхода из `recovery` и перезапускает службу сервера 1С (`rphost`)
- 3 Старый `primary` возвращаем в строй как реплику через `pg_rewind --source-server` — для этого заранее включён `wal_log_hints=on`
- 4 Split-brain исключаем: `nopreempt` + второй `track_script` пингует шлюз, изолированная нода сама снимает VIP; назад VIP возвращаем только руками после проверки
- 5 Каждое переключение — алерт в Telegram дежурному инженеру и письмо ответственному со стороны заказчика

## РЕЗУЛЬТАТ

Простой пользователь при отказе `primary` — десятки секунд вместо часов восстановления из бэкапа; плановые работы на железе (прошивки, диски) проходим переключением без остановки учёта.

## КЛЮЧЕВОЙ НЮАНС

`pg_rewind` требует `wal_log_hints=on` либо контрольных сумм при `initdb` — включаем на этапе развёртывания, иначе возврат старого `primary` = полный `pg_basebackup` на десятки гигабайт и часы ожидания.

# Тюнинг под 1С и мониторинг репликации в Zabbix

## Что настраиваем

postgresql.conf по приложению Postgres Pro для решений 1С + шаблон PostgreSQL by Zabbix agent 2

## Как мы это делаем

- 1 shared\_preload\_libraries='online\_analyze, plantuner';  
plantuner.fix\_empty\_table='on' чинит планы по свежесозданным пустым таблицам, online\_analyze.table\_type='temporary'
- 2 standard\_conforming\_strings=off и escape\_string\_warning=off — обязательные для 1С значения из официального приложения документации Postgres Pro
- 3 temp\_buffers 32-256MB и max\_locks\_per\_transaction=256 — 1С массово строит временные таблицы; shared\_buffers ~25% RAM, effective\_cache\_size — не менее половины RAM
- 4 Zabbix agent 2 с плагином PostgreSQL: postgresql.replication.lag.sec, роль ноды, статус стриминга; пороги дисков 80% warning / 90% critical
- 5 fsync=on всегда: «ускорение» его отключением при сбое питания превращает базу в невозстановимую

## РЕЗУЛЬТАТ

Отчёты 1С не деградируют после массовых загрузок документов, а дежурный видит рост лага репликации и заполнение дисков до того, как они превратятся в инцидент с простоем.

## КЛЮЧЕВОЙ НЮАНС

Приложение Postgres Pro не советует online\_analyze современным версиям 1С — платформа сама обновляет статистику временных таблиц. Включаем модуль, только если планировщик ошибается в оценке строк.

# Подводные камни

## ✗ Реплика вместо бэкапа

Ошибка бухгалтера и шифровальщик реплицируются на standby мгновенно. Держим ночной pg\_basebackup на NAS, недельный pg\_dump и офлайн-копию в сейфе.

## ✗ synchronous\_commit без имени реплики

Пустой synchronous\_standby\_names молча выключает синхронность — коммит подтверждается локально. Контролируем sync\_state=sync в pg\_stat\_replication.

## ✗ Реплика «отваливается» после ребута

Без слота primary успевает удалить нужные WAL. Создаём физический слот (pg\_basebackup -C --slot) и мониторим его, чтобы не переполнить pg\_wal.

## ✗ pg\_rewind не запускается

Требуется wal\_log\_hints=on или контрольных сумм (initdb --data-checksums). Включаем при развёртывании — на живом кластере это рестарт и конвертация.

## ✗ Ванильный PostgreSQL без патчей 1C

Платформа поддерживает только сборки с патчами 1C (например 17.9-3.1C) или линейку Postgres Pro 1C — без них ошибки на сложных запросах и кривые планы.

## ✗ Split-brain при мигающей сети

Два primary одновременно — расхождение данных. Ставим porgram, в track\_script добавляем пинг шлюза (изолированная нода снимает VIP), назад VIP возвращаем только вручную.

## ✗ 1C:Сервер не видит новый primary

gphost держит старые соединения к СУБД и сыплет ошибками. В notify\_master-скрипт добавляем рестарт службы сервера 1C сразу после pg\_ctl promote.

## ✗ Отставание на закрытии месяца

Массовые проводки раздувают WAL быстрее, чем отдаёт сеть. Выделенный линк под репликацию, wal\_keep\_size 8-16GB и алерт на рост replay\_lag.



# Как правильно

## МИНИМУМ

- Сборка PostgreSQL с патчами 1C (17.x) вместо ванильной, fsync=on без исключений
- Ежедневный pg\_basebackup на отдельный NAS + еженедельный логический pg\_dump
- Zabbix-контроль дисков, процессов postgres и rphost с алертами в Telegram

## НОРМАЛЬНО

- Вторая нода со streaming replication и физическим слотом репликации
- keepalived с VIP: pg\_isready каждые 2 с, автоматический failover без админа
- synchronous\_commit=on с явным synchronous\_standby\_names — RPO ноль
- Мониторинг лага репликации (pgsql.replication.lag.sec) с порогами и алертами

## ХОРОШО

- Выделенный 10GbE-линк под репликацию: латентность <1 мс, синхронность без тормозов
- wal\_log\_hints=on + pg\_rewind: возврат старой ноды без полной пересинхронизации
- Пинг третьей точки (шлюза) в track\_script против split-brain + квартальные учения по failover с замером простоя
- Офлайн-копия бэкапа вне контура (диск в сейфе) — защита от шифровальщика

# Чек-лист самопроверки

---

- СУБД — сборка с патчами 1C (releases.1c.ru или Postgres Pro 1C), версия есть в списке поддерживаемых v8.1c.ru?
- Есть ли вторая нода PostgreSQL и подтверждён ли `sync_state=sync` в `pg_stat_replication`?
- Переживёт ли офис отказ сервера СУБД без участия админа — и за сколько секунд?
- Создан ли физический слот репликации и мониторится ли отставание реплики в секундах?
- Включён ли `wal_log_hints=on` — сможете вернуть старый `primary` через `pg_rewind` без полной копии?
- Бэкапы идут на отдельное железо, а не на реплику? Есть офлайн-копия вне контура?
- Проводилось ли тестовое восстановление с проверкой базы через «Тестирование и исправление»?
- Загружен ли `plantuner` (`fix_empty_table=on`), выставлены ли `standard_conforming_strings=off` и `escape_string_warning=off`?
- Придёт ли алерт дежурному в Telegram при росте лага репликации и заполнении диска на 80%?
- Проводите ли плановые учения по failover хотя бы раз в квартал с замером времени простоя?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



# Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Аудит текущего сервера 1С/СУБД и проект отказоустойчивой пары под ваш объём баз и число рабочих мест
- Миграция с MS SQL на PostgreSQL/Postgres Pro 1С с обкаткой на копии баз и замером скорости до/после
- Развёртывание репликации, keepalived-failover, схемы бэкапов и Zabbix-мониторинга под ключ
- Регламентные учения по переключению и квартальные тестовые восстановления с отчётом руководителю
- Поддержка по SLA: дежурный инженер реагирует на алерты кластера СУБД круглосуточно

**15+**

лет в ИТ-поддержке

**50**

рабочих мест — наш профиль

**МТС**

дата-центр, Москва

## КОНТАКТЫ

# Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh\_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru

# Техническая база

---

- 01** High Availability, Load Balancing, and Replication — документация (postgresql.org — 17)
- 02** pg\_basebackup, pg\_rewind, pg\_ctl — Reference (postgresql.org — 17)
- 03** Приложение «Настройка Postgres Pro для решений 1С» (postgrespro.ru — 17)
- 04** Системные требования «1С:Предприятия 8» — СУБД PostgreSQL (v8.1c.ru — 8.3.27)
- 05** Бесплатные сборки Postgres Pro для 1С (1c.postgres.ru — 2026)
- 06** Шаблон «PostgreSQL by Zabbix agent 2» (zabbix.com — 7.0 LTS)
- 07** keepalived.conf — vrrp\_script / track\_script / notify (keepalived.org — 2.x)
- 08** Наш шаблон promote\_standby.sh + регламент возврата ноды (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

