Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

PostgreSQL 17: миграция боевых кластеров без даунтайма

Наша методика перевода кластеров PG 13–16 на ветку 17: pg_upgrade, логическая репликация, бэка…

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Клиенты работают на PostgreSQL 13–16 под 1С, CRM и веб-системами: ветка 13 уже без патчей безопасности, полные ночные бэкапы съедают диски и окна, VACUUM на больших таблицах не успевает. Обновление откладывают из страха простоя. Мы переводим такие кластеры на PG 17 по отработанному регламенту: простой от секунд до 10 минут, инкрементальные бэкапы и план отката на каждом шаге.

Почему это важно бизнесу

- EOL-ветка без security-патчей — прямой риск для БД бухгалтерии и персданных; ФЗ-152 и договоры контрагентов этого не прощают
- Полный бэкап каждую ночь — терабайты диска и часы окна; инкременты PG 17 режут объём и время на порядок
- Незапланированный простой БД останавливает 1С, кассы и отгрузки; плановая миграция стоит минуты, авария — дни
- Быстрый VACUUM и планировщик PG 17: те же серверы тянут больше сессий, апгрейд железа откладывается

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

17.10

актуальный минор ветки 17 — ставим только его: минорные патчи закрывают CVE без смены формата…

PostgreSQL 17, релиз 05.2026

до 20x

меньше памяти на dead-tuples у VACUUM (структура TidStore); снят скрытый потолок 1 ГБ для main…

по докам PostgreSQL 17

10 дней

wal_summary_keep_time по умолчанию — глубина WAL-сводок для pg_basebackup --incremental; мы ст…

по докам PostgreSQL 17

2–10 мин

даунтайм при pg_upgrade --link --jobs: переносятся хардлинки, а не данные, объём БД почти не в…

наш стандарт

11.2029

конец поддержки ветки 17 — запас 3+ года; ветка 13 снята с поддержки в ноябре 2025

versioning policy postgresql.org

30 с

порог алерта replication lag на время миграции логической репликацией и на боевых репликах

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Миграция pg_upgrade --link на том же хосте

Что настраиваем

Standalone-серверы и базы 1С до ~1 ТБ, где допустимо окно 10 минут; узел БД, PG 13–16 → 17

Как мы это делаем

1. Ставим PG 17 рядом (postgresql-17), переносим нестандартные GUC из старых postgresql.conf/pg_hba.conf, ставим сборки всех расширений под 17
2. Заранее гоняем pg_upgrade --check — он работает без остановки старого кластера и показывает несовместимости до окна
3. Окно: systemctl stop postgresql@15-main → pg_upgrade --link --jobs=<число ядер> → start 17-main; порт и точки подключения приложений не меняем
4. Сразу vacuumdb --all --analyze-in-stages: статистика планировщика при апгрейде не переносится
5. Сутки наблюдаем pg_stat_statements и логи, и только затем запускаем delete_old_cluster.sh

РЕЗУЛЬТАТ

Типовое окно 2–10 минут вместо часов pg_dump/pg_restore; старый каталог остаётся точкой отката до первого старта нового кластера, поэтому регламент фиксирует момент невозврата и все проверки до него.

КЛЮЧЕВОЙ НЮАНС

--link создаёт хардлинки: после первого запуска PG 17 старый кластер использовать нельзя. Полный бэкап перед окном обязателен, а --check прогоняем на staging-клоне с боевым набором расширений.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Zero-downtime через логическую репликацию

Что настраиваем

Базы 24/7 — интернет-магазины, кассовые и медицинские бэкенды, где даже 10 минут простоя недопустимы

Как мы это делаем

1. На источнике: wal_level=logical, CREATE PUBLICATION FOR ALL TABLES; помним — DDL и sequences логическая репликация не переносит
2. На приёмнике PG 17: схема через pg_dump --schema-only, затем CREATE SUBSCRIPTION; начальная синхронизация идёт без остановки источника
3. Мониторим pg_stat_subscription, ждём received_lsn ≈ latest_end_lsn; лаг держим под порогом 30 с
4. Переключение: стоп записи, выравнивание LSN, setval() всех sequences, перевод приложения, DROP SUBSCRIPTION — простой считаные секунды
5. После переезда включаем failover-слоты PG 17: failover=true у подписки + sync_replication_slots=on на standby — слот переживает switchover

РЕЗУЛЬТАТ

Простой измеряется секундами при любом объёме БД; заодно получаем физически «чистую» копию без накопленного bloat — база после переезда компактнее и заметно быстрее на тех же запросах.

КЛЮЧЕВОЙ НЮАНС

Слоты удерживают WAL: брошенная подписка забивает диск источника. Всегда ставим max_slot_wal_keep_size, алерт на inactive-слоты в pg_replication_slots, подписку удаляем сразу после переключения.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Инкрементальные бэкапы вместо ежедневных full

Что настраиваем

Контур резервного копирования PG 17: узел БД + внешнее бэкап-хранилище (NAS/FTP)

Как мы это делаем

1. Включаем summarize_wal=on (достаточно reload), wal_summary_keep_time = 14d — с запасом больше интервала между полными бэкапами
2. Полный pg_basebackup по воскресеньям с --checkpoint=fast --manifest-checksums=SHA256; в будни — --incremental=<backup_manifest предыдущего>
3. Восстановление: pg_combinebackup full incr1..incrN -o <каталог>, дальше как обычный basebackup; WAL-архив через archive_command даёт PITR
4. Ежемесячный автоматический restore цепочки на стенде + pg_verifybackup; результат уходит отчётом в Telegram-мониторинг

РЕЗУЛЬТАТ

Недельная цепочка занимает доли объёма семи полных копий, ночное окно копирования сокращается в разы, а вместе с WAL-архивом получаем восстановление на любую точку недели, а не только на момент бэкапа.

КЛЮЧЕВОЙ НЮАНС

Инкремент строится по манифесту предыдущего бэкапа: потеря одного звена рвёт цепочку — тогда немедленно новый full. Наличие сводок контролируем через pg_available_wal_summaries().

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Нет ANALYZE после pg_upgrade**
  Статистика планировщика не переносится — запросы деградируют в разы. Сразу после старта: vacuumdb --all --analyze-in-stages.

- ✕ **Расширения без сборки под 17**
  pg_upgrade падает или кластер не стартует. До окна ставим пакеты PostGIS 3.5+/TimescaleDB 2.17+ под PG 17 и прогоняем --check.

- ✕ **Ванильный PostgreSQL под 1С**
  Платформа 1С требует патченной сборки (1С/Postgres Pro) — на ванили ошибки блокировок. Базы 1С мигрируем только на профильные сборки.

- ✕ **--link без свежего бэкапа**
  После первого старта PG 17 старый кластер уже не поднять — файлы общие через хардлинки. Полный бэкап и фиксация точки невозврата — до окна.

- ✕ **Потерянные слоты репликации**
  При апгрейде НА 17 логические слоты не переносятся (перенос pg_upgrade работает только со старого кластера 17+). Подписчиков пересоздаём по плану.

- ✕ **Разрыв цепочки инкрементов**
  Если WAL-сводки удалены раньше очередного бэкапа, инкремент не построится. Держим wal_summary_keep_time больше интервала и мониторим сводки.

- ✕ **WAL-распухание от брошенного слота**
  Неактивный слот держит WAL до заполнения диска. Ставим max_slot_wal_keep_size и алерт на inactive-слоты в pg_replication_slots.

- ✕ **pg_dump как единственный бэкап**
  Логический дамп не даёт PITR и восстанавливается часами. Основной контур — pg_basebackup + WAL-архив; дамп только как дополнение.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Уйти с EOL-веток (13 и старше); минорные патчи — ежеквартально
- pg_upgrade --check на клоне со всеми боевыми расширениями до окна
- Полный pg_basebackup + тестовый restore перед любой миграцией

НОРМАЛЬНО

- pg_upgrade --link --jobs + регламент с зафиксированной точкой невозврата
- vacuumdb --analyze-in-stages и сутки контроля pg_stat_statements после окна
- Инкрементальные бэкапы: summarize_wal=on, full в вс + incremental по будням
- postgres_exporter/Zabbix: replication lag, autovacuum, pg_stat_checkpointer

ХОРОШО

- Zero-downtime миграция логической репликацией с переключением по LSN
- Failover-слоты: failover=true + sync_replication_slots=on на standby
- Ежемесячный авто-restore pg_combinebackup + pg_verifybackup на стенде
- transaction_timeout против зависших транзакций, блокирующих VACUUM

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Ветка PostgreSQL поддерживается и стоит последний минорный релиз (для 17 — 17.10)?
- [ ] Есть свежий полный бэкап и подтверждённый тестовый restore перед окном миграции?
- [ ] pg_upgrade --check пройден на staging-клоне со всеми боевыми расширениями?
- [ ] Все расширения (PostGIS, TimescaleDB, pg_stat_statements) имеют сборки под 17?
- [ ] После апгрейда выполнен vacuumdb --all --analyze-in-stages?
- [ ] Включён summarize_wal и цепочка инкрементов реально восстанавливалась pg_combinebackup?
- [ ] Мониторинг ловит replication lag, распухание WAL от слотов и застой autovacuum?
- [ ] Для баз 1С используется патченная сборка PostgreSQL, а не ванильная?
- [ ] Точка невозврата --link и план отката зафиксированы в регламенте окна?
- [ ] Драйверы приложений (libpq, JDBC, psycopg) проверены на совместимость с 17?

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

ЧТО МЫ ДЕЛАЕМ

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 кластера: версии, расширения, bloat, готовность к PG 17 — отчёт с планом миграции
- Миграция под ключ: staging-прогон, pg_upgrade --link или логическая репликация без простоя
- Инкрементальные бэкапы PG 17 + регламент ежемесячных тестовых восстановлений
- Мониторинг PostgreSQL: postgres_exporter/Zabbix, алерты lag/autovacuum/checkpoint
- Перевод 1С с MS SQL на PostgreSQL с тюнингом патченной сборки под платформу

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

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Release 17 — Appendix E. Release Notes (postgresql.org — 17)

02 pg_upgrade — PostgreSQL Server Applications (postgresql.org — 17)

03 pg_basebackup / pg_combinebackup / pg_verifybackup (postgresql.org — 17)

04 Logical Replication — Failover Slots, Subscription (postgresql.org — 17)

05 WAL Configuration — summarize_wal, wal_summary_keep_time (postgresql.org — 17)

06 Versioning Policy — сроки поддержки веток (postgresql.org — 2026)

07 Наш регламент миграционного окна БД (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.