



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

BorgBackup для Linux-серверов офиса: наша схема 3-2-1

Дедуплицированный шифрованный бэкап серверов
1C/PostgreSQL, Samba и nginx: Borg 1.4 + borgmatic

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Типовой Linux-сервер клиента несёт сразу 1С на PostgreSQL, шары Samba и веб-сервисы, а бэкап — самописный cron-скрипт, который молча падает и не хранит историю. Отказ RAID или шифровальщик оставляют компанию без базы и первички — недели ручного ввода. Мы закрываем это связкой Borg 1.4 + borgmatic 2.1 по схеме 3-2-1 с append-only копией вне офиса.

Почему это важно бизнесу

- Потеря базы 1С без проверенного бэкапа — недели ручного восстановления документов силами бухгалтерии вместо часов штатного restore
- Шифровальщики первым делом ищут доступные по сети бэкапы; без append-only копии вне офиса выбор — платить вымогателю или терять данные
- Дедупликация сокращает хранилище в разы: год истории помещается в 2-4 объёма исходных данных — меньше дисков NAS и трафика в ЦОД
- Шифрование на стороне клиента (repokey, AES) позволяет держать копию в стороннем ЦОД без риска утечки персональных данных



Ключевые параметры реализации

1.4.5

Ветка Borg 1.4 — наш прод-стандарт (релиз 19.07.2026); Borg 2.0 пока в бете, в бой не ставим

borgbackup.org, releases

2 MiB

Целевой блок дедупликации: buzhash-чанкер с параметрами 19,23,21,4095 по умолчанию

доки Borg: internals

zstd,9

Компромисс сжатия: заметно плотнее lz4 и в разы быстрее lzma на дампах 1C и файловых шарах

наш стандарт

14/8/12

Политика prune (daily/weekly/monthly): история около года; следом обязательный borg compact

наш стандарт

--last 3

Еженедельный borg check --archives-only по трём последним архивам — ловим порчу репозитория ра...

доки Borg: borg check

90 дн

Интервал restore-теста: подъём последнего архива в чистой VM с запуском PostgreSQL и 1C

наш стандарт



Репозиторий и шифрование: узел-источник + NAS

Что настраиваем

Офисный Linux-сервер (Debian 12/Ubuntu 24.04): 1С на PostgreSQL, шары Samba; горячая копия — NAS в /mnt/backup по NFS

Как мы это делаем

- 1 Ставим borgbackup 1.4.x: штатный apt Debian 12/Ubuntu 24.04 даёт только ветку 1.2 (1.2.4/1.2.8), поэтому берём bookworm-backports или официальный standalone-бинарь;...
- 2 borg init --encryption=repokey-blake2: ключ лежит в репозитории под passphrase, BLAKE2b-256 быстрее SHA-256 на CPU без аппаратного SHA
- 3 Passphrase 32+ символа — в менеджер паролей клиента и наш, бумажный конверт в сейф; отдельно делаем borg key export на защищённый носитель
- 4 Ночной borg create --stats --show-rc --compression zstd,9 --exclude-caches --one-file-system: /etc, /home, /srv, /opt, /usr/local + свежие дампы БД
- 5 borg prune --keep-daily 14 --keep-weekly 8 --keep-monthly 12 и сразу borg compact — иначе место на NAS не освобождается

РЕЗУЛЬТАТ

Год истории архивов при хранилище в 2–4 объёма исходных данных за счёт блочной дедупликации (~2 MiB чанки); восстановление одного файла — borg mount через FUSE и обычный cp, без полного extract

КЛЮЧЕВОЙ НЮАНС

repokey хранит ключ внутри репозитория: потеря passphrase равна потере данных, поэтому borg key export — обязательный пункт чек-листа внедрения, а не опция



Консистентные дампы PostgreSQL и borgmatic

Что настраиваем

Сервер 1С:Предприятие на PostgreSQL; оркестрация бэкапа через borgmatic 2.1 (YAML-конфиг, systemd-таймер)

Как мы это делаем

- 1 /var/lib/postgresql исключаем из архива полностью: файлы работающего кластера неконсистентны и не восстанавливаются
- 2 Хук postgresql_databases в borgmatic: pg_dump формата custom стримится прямо в архив через именованный канал — без временных дампов на диске
- 3 retention: keep_daily 14 / keep_weekly 8 / keep_monthly 12; проверки checks: repository + archives по расписанию borgmatic
- 4 systemd borgmatic.timer на 02:30 с RandomizedDelaySec; итог прогона уходит пингом в healthchecks и алертом в Telegram при ошибке

РЕЗУЛЬТАТ

Восстановление отдельной базы 1С — pg_restore из custom-дампа прямо из архива; штатному админу клиента достаточно читать один YAML вместо развесистого bash, конфиг проверяется borgmatic config validate

КЛЮЧЕВОЙ НЮАНС

С borgmatic 2.0 формат command-хуков изменён против 1.x (before_*/after_* заменены на commands:): старые конфиги прогоняем через borgmatic config generate --source, а не правим вручную



Внеофисная копия с защитой от шифровальщика

Что настраиваем

Реплика репозитория в другом ЦОД по SSH (borg serve); третья, офлайн-копия — отчуждаемый USB-диск раз в месяц

Как мы это делаем

- 1 Второй репозиторий `ssh://borg@host/./srv01`: Borg работает клиент-сервер через `borg serve`, на хранилище нужен только SSH и сам `borg`
- 2 В `authorized_keys`: `command="borg serve --restrict-to-repository /storage/srv01 --append-only",restrict` — ключ с сервера умеет только дописывать
- 3 `prune` и `compact` этой копии — только с доверенного узла хранения отдельным ключом без `append-only`, по своему расписанию
- 4 Ежемесячно на стороне хранилища `borg check --verify-data` (полная проверка содержимого чанков), на источнике еженедельно `--archives-only --last 3`
- 5 Раз в квартал — `restore`-тест: `extract` последнего архива в чистую VM, подъём PostgreSQL из дампа, вход в 1С и сверка контрольных отчётов

РЕЗУЛЬТАТ

Компрометация офисного сервера не уничтожает историю: ключ клиента физически не может удалить архивы, а удаления в `append-only` режиме остаются в журнале транзакций до `compact` и обратимы

КЛЮЧЕВОЙ НЮАНС

`append-only` работает, только если закреплён на принимающей стороне: тот же ключ с правом `borg delete/prune` обесценивает схему — права разводим по разным SSH-ключам



Подводные камни

✗ Бэкап живых файлов PostgreSQL

Копирование `/var/lib/postgresql` на работающей базе даёт неконсистентные файлы, которые не поднимутся; только `pg_dump/pg_dumpall` или `pg_basebackup`

✗ Passphrase в одном экземпляре

`herokey` шифрует ключ парольной фразой: без пароля репозиторий неотличим от шума. Храним в двух хранилищах + бумага, плюс `borg key export`

✗ prune без compact

С Borg 1.2+ `prune` лишь помечает сегменты к удалению — место возвращает только `borg compact`; без него NAS тихо заполняется до отказа бэкапов

✗ append-only только на клиенте

Флаг должен быть зажат в `command=` в `authorized_keys` принимающей стороны; иначе украденный ключ выполнит `borg delete` и снесёт историю

✗ borg check как полная гарантия

Быстрый `check` проверяет структуру сегментов, но не содержимое чанков: раз в месяц нужен `--verify-data` и реальный `restore`-тест в VM

✗ Несколько хостов в один репозиторий

Каждый клиент держит свой кэш чанков; чужие архивы вынуждают полный `resync` кэша и многочасовые бэкапы. Правило: один источник — один репозиторий

✗ Тишина cron = успех

Скрипт может месяцами падать молча. Смотрим `rs` через `--show-rs`, шлём итог в `healthchecks/Telegram` и алертим по возрасту последнего архива

✗ rclone-синк репозитория во время create

Копирование репозитория посреди транзакции даёт неконсистентную реплику; синхронизируем после завершения `create/compact` или ведём второй репозиторий



Как правильно

МИНИМУМ

- Borg 1.4.x (backports/standalone, в apt Debian 12/Ubuntu 24.04 — только 1.2), репозитории...
- Дампы `pg_dump -Fc` вместо файлов кластера; `/var/lib/postgresql` — в `exclude`
- Еженедельный `borg check --archives-only --last 3` с алертом при `rc != 0`
- `Passphrase` в двух хранилищах + `borg key export` на отдельный носитель

НОРМАЛЬНО

- `borgmatic 2.1`: YAML-конфиг, хук `postgresql_databases`, `systemd`-таймер
- Вторая копия по SSH в другой ЦОД через `borg serve` на выделенный ключ
- Контроль возраста последнего архива и кода возврата в `Zabbix/healthchecks`
- `borg compact` по расписанию и контроль свободного места на NAS

ХОРОШО

- `append-only` в `authorized_keys`; `prune/compact` — только с узла хранения
- Ежемесячный `borg check --verify-data` на стороне хранилища
- Квартальный `restore`-тест в VM: PostgreSQL из дампа + вход в 1C
- Третья офлайн-копия на отчуждаемом диске с ротацией вне офиса



Чек-лист самопроверки

- | | |
|--|---|
| ☐ Исключены ли файлы работающей СУБД из архива и настроен ли консистентный дамп (<code>pg_dump -Fc</code>)? | ☐ Приходит ли алерт при ненулевом гс и при отсутствии свежего архива за последние сутки? |
| ☐ Хранится ли passphrase минимум в двух независимых местах и сделан ли <code>borg key export</code> ? | ☐ Выполняется ли <code>borg check</code> еженедельно, а полный <code>--verify-data</code> — хотя бы раз в месяц? |
| ☐ Есть ли копия вне офиса, которую нельзя удалить ключом с бэкапируемого сервера (<code>append-only</code>)? | ☐ Был ли за последние 90 дней <code>restore-тест</code> с подъёмом PostgreSQL и входом в 1С в отдельной VM? |
| ☐ Запускается ли <code>borg compact</code> после <code>prune</code> и хватает ли на NAS места на 2–4 объёма данных? | ☐ Зафиксированы ли RTO/RPO в регламенте и знает ли ответственный порядок <code>bare-metal restore</code> ? |

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 текущих бэкапов Linux-серверов: что реально восстановимо и где сбои происходят молча
- Внедрение Borg 1.4 + borgmatic 2.1 под ключ за один день: репозитории, шифрование, дампы БД, таймеры
- Внеофисная append-only копия на нашем хранилище в ЦОД с отдельными SSH-ключами
- Мониторинг бэкапов (Zabbix, Telegram, healthchecks) и квартальные restore-тесты с протоколом
- Аварийное восстановление: bare-metal restore сервера и подъём 1C/PostgreSQL из дампов

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** Borg Documentation: Quick Start, borg create/prune/compact (borgbackup.readthedocs.io — 1.4.5)
- 02** Borg Internals: chunker и формат репозитория (borgbackup.readthedocs.io — 1.4)
- 03** Borg Security & Encryption: режимы reprokey/keyfile, BLAKE2b (borgbackup.readthedocs.io — 1.4)
- 04** Borg FAQ / borg serve: append-only mode (borgbackup.readthedocs.io — 1.4)
- 05** borgmatic Reference: configuration, database hooks, config generate (torsion.org — 2.1)
- 06** PostgreSQL Docs: pg_dump, pg_dumpall, pg_restore (postgresql.org — 16)
- 07** Шаблон ITfresh: регламент 3-2-1 и restore-тестов (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

