



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Резервное копирование 3-2-1 для малого офиса: настройка с нуля

Наша схема на open source: rsync-снимки, дедупликация Borg/Restic, офсайт в S3 с Object Lock

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Один сбой RAID, шифровальщик или ошибочное удаление — и офис до 50 рабочих мест встаёт: без 1С не идут платежи и отгрузки, без файловых шар не работают бухгалтерия и юристы. Мы разворачиваем 3-2-1 на open-source-стеке (rsync + Borg/Restic + S3) за два дня: три копии, два типа носителей, одна вне офиса — с иммутабельностью, которую не сотрёт даже взломанный сервер.

Почему это важно бизнесу

- Простой без 1С и файловых шар — остановленные платежи, отгрузки и зарплата; день простоя дороже всей системы бэкапа
- Современный шифровальщик первым делом ищет и стирает бэкапы — копия на сетевой шаре с правами записи не защищает
- Стек полностью open source: затраты — только NAS, диски и ~600-1000 ₽/мес за S3, без коммерческих лицензий
- Локальная копия даёт RTO ~30 минут; тянуть 500 ГБ только из облака по каналу 100 Мбит — 12-15 часов простоя
- Проверенное по протоколу восстановление превращает бэкап из «надежды» в страховку с измеренным RTO



Ключевые параметры реализации

1.4.5

BorgBackup на NAS: repokey-blake2, сжатие zstd,9, приём только через borg serve --append-only

Borg 1.4, borgbackup.readthedocs.io

0.19.1

Restic для офсайта: нативный S3-бэкенд, клиентское шифрование AES-256, forget/prune по схеме G...

restic.readthedocs.io

n/10

--read-data-subset у restic check: детерминированные десятые доли pack-файлов, весь репозитори...

restic.readthedocs.io, 0.19

26 ч

Порог Zabbix-триггера на возраст последнего успешного бэкапа: сутки + 2 часа запаса на окно

наш стандарт, Zabbix 7.0 LTS

7/4/6/2

GFS-етенция
daily/weekly/monthly/yearly: откат до двух лет назад без раздувания хранилища

наш стандарт (borg prune)

2 дня

Срок внедрения всех трёх уровней под ключ в офисе до 50 PM, включая тестовое восстановление

наш стандарт



Уровни 1-2: снимки rsync и Borg-репозиторий на NAS

Что настраиваем

Файловый сервер (Samba/Windows-шары) и сервер 1С → NAS в серверной клиента; локальная часть схемы 3-2-1

Как мы это делаем

- 1 Инвентаризация: критичные (1С, шары бухгалтерии; RPO 1 ч), важные (RPO 24 ч), холодные (RPO 7 дней) — от класса зависят расписание и глубина хранения
- 2 `rsync -aHAXx --numeric-ids --delete --link-dest=daily.1 → daily.0`: неизменённые файлы становятся хардлинками, 7 ежедневных снимков занимают почти как один
- 3 Запуск через `systemd timer`, не `cron`: `Type=oneshot, Nice=15, IOSchedulingClass=best-effort prio 7`, весь вывод — в `journald`
- 4 Borg 1.4 на NAS: `borg init --encryption=repokey-blake2, borg create --compression zstd,9` с исключением `*.tmp` и `*.log`
- 5 Ротация `borg prune` по GFS: `--keep-daily 7 --keep-weekly 4 --keep-monthly 6 --keep-yearly 2`

РЕЗУЛЬТАТ

Восстановление файла или каталога — минуты: `daily.N` выглядит как обычная папка, права и `mtime` сохранены. Borg-репозиторий зашифрован и дедуплицирован, история до двух лет не раздувает NAS, а откат возможен даже если инцидент заметили спустя месяцы.

КЛЮЧЕВОЙ НЮАНС

`Nice=15` и `IOSchedulingPriority=7` обязательны: `rsync -H` на шарах с миллионами хардлинков без них парализует дисковую подсистему; ключ Borg сразу после `init` выгружаем через `borg key export` и храним вне сервера.



Уровень 3: офсайт в S3 с иммутабельностью

Что настраиваем

Restic 0.19 → S3-совместимый бакет (Yandex Object Storage / MinIO) в другом ЦОД; отдельный бакет и ключи на клиента

Как мы это делаем

- 1 restic init на s3:-репозиторий; пароль в RESTIC_PASSWORD_FILE, дубли ключей — в KeePassXC у клиента и в нашем защищённом сейфе
- 2 Выгрузка баз 1С в *.dt в 00:00, затем restic backup --tag с --exclude-file; критичные данные уходят в S3 ежедневно в 03:00, вне рабочего окна канала
- 3 Object Lock на бакете в режиме compliance на 14 дней: версии объектов не удалить даже с валидными ключами доступа
- 4 Сервисный аккаунт клиента без DeleteObject (только Put/Get/List), консоль провайдера — строго с MFA
- 5 restic forget --keep-daily 14 --keep-weekly 4 --keep-monthly 6 --prune — с отдельного хоста управления, не с клиентского сервера

РЕЗУЛЬТАТ

Шифровальщик с правами на клиентском сервере не может уничтожить офсайт: ключей на удаление там нет, Object Lock держит версии заданный срок. Данные у провайдера лежат зашифрованными на стороне клиента — компрометация хранилища не раскрывает содержимое.

КЛЮЧЕВОЙ НЮАНС

Prune и Object Lock конфликтуют: срок блокировки не должен превышать минимальную ретенцию forget (у нас 14 дней = --keep-daily 14), иначе prune падает на защищённых объектах — связку обкатываем на тестовом бак...



Мониторинг и регламент тестовых восстановлений

Что настраиваем

Zabbix 7.0 LTS (zabbix_agent2) на каждом бэкап-хосте + ежемесячный протокол восстановления дежурного инженера

Как мы это делаем

- 1 UserParameter backup.last[*] читает mtime маркера .success после каждого задания; триггер — при отставании более 26 ч (сутки + 2 ч на окно)
- 2 Контроль размера через borg info --json (.cache.stats.unique_csize) — ловим «пустые» бэкапы после переименования путей
- 3 restic check --read-data-subset=n/10 еженедельно (n — номер недели цикла): за 10 прогонов репозиторий проверяется чтением реальных данных, а не только метаданных
- 4 Ежемесячно — восстановление файла (права + mtime) и каталога на тестовый хост; ежеквартально — подъём *.dt в чистом кластере 1C
- 5 Алерты Zabbix → Telegram дежурному, сводный отчёт по всем заданиям — на почту в 09:00

РЕЗУЛЬТАТ

«Тихая смерть» бэкапа исключена: о сбое узнаём утром того же дня, а не в день аварии. Протоколы восстановлений дают измеренный, а не предполагаемый RTO: файл — минуты, база 1C — до 2 часов, полный хост — в пределах регламента.

КЛЮЧЕВОЙ НЮАНС

Мониторим не факт запуска, а факт успеха: маркер .success пишется только при нулевом коде выхода всего пайплайна (выгрузка → бэкап → prune) — иначе триггер молчит про сломанную середину цепочки.



Подводные камни

✗ **NAS в соседней комнате как «офсайт»**

Пожар или потоп накрывает обе копии разом; офсайт — только другое здание: S3-бакет или VPS в другом ЦОД с шифрованием на стороне клиента

✗ **Бэкап-репозиторий доступен по SMB из домена**

Шифровальщик стирает его первым; у нас NAS принимает только SSH с `command="borg serve --append-only"`, без `pty` и форвардинга

✗ **Диски одной партии на обоих уровнях**

Массив из одной партии сыплется синхронно; второй уровень — другая модель/партия дисков или иная среда: S3, отдельный NAS

✗ **Бэкап живых файлов 1С вместо выгрузки**

Файлы работающей СУБД/1С копируются неконсистентно; сначала выгрузка в `*.dt` (или `VSS/pg_dump`), затем Borg/Restic поверх выгрузки

✗ **Ключ шифрования только на сервере-источнике**

Погиб сервер — погиб и доступ к бэкапу; ключ Borg (`borg key export`) и пароль Restic дублируем в KeePassXC клиента и нашем сейфе

✗ **Prune с того же хоста, что и backup**

Скомпрометированный сервер вычистит историю; удаление и `forget` — с отдельного хоста управления под отдельным ключом

✗ **Проверка = «задание отработало без ошибок»**

Код 0 не гарантирует читаемость: еженедельно `restic check --read-data-subset` / `borg check`, ежемесячно — тестовое восстановление

✗ **cron вместо systemd timer**

Ошибки теряются; `oneshot`-юнит пишет в `journald`, а `OnCalendar` с `Persistent=true` догоняет пропущенные запуски после ребута



Как правильно

МИНИМУМ

- `rsync --link-dest` на NAS ежедневно, 7 снимков, `systemd timer` с логом в `journald`
- Restic в S3 для критичных данных раз в сутки, шифрование на стороне клиента
- Zabbix-триггер на возраст бэкапа >26 ч + ежедневный отчёт на почту
- Выгрузка 1C в *.dt перед бэкапом, пароли репозитория — в KeePassXC

НОРМАЛЬНО

- Borg 1.4 на NAS: `repokey-blake2`, `zstd`, ретенция GFS 7/4/6/2, `append-only` по SSH
- Object Lock на S3-бакете, сервисный аккаунт без права `DeleteObject`
- `restic check --read-data-subset=n/10` еженедельно, `borg check` ежемесячно
- Ежемесячное тестовое восстановление файла и каталога по протоколу

ХОРОШО

- Prune/forget вынесены на отдельный хост управления с собственным ключом
- Ежеквартально полный прогон DR: *.dt в чистом кластере 1C на запасном хосте
- Раздельные бакеты и S3-ключи на каждую систему, MFA на консоль провайдера
- Печатная «Процедура восстановления» в серверной + копия у руководителя



Чек-лист самопроверки

- ☐ Все три уровня отработали минимум три ночи подряд без ошибок в journald и Zabbix?
- ☐ Офсайт-копия физически в другом здании или ЦОД, а не на NAS в соседней комнате?
- ☐ Бэкап-учётка вне доменных админ-групп, репозиторий недоступен по SMB на запись?
- ☐ Включены append-only на Borg и Object Lock на S3-бакете для критичных данных?
- ☐ Ключи Borg и пароли Restic продублированы вне сервера-источника (KeePassXC)?

- ☐ Базы 1С/СУБД выгружаются в консистентный дамп (*.dt, pg_dump) перед файловым бэкапом?
- ☐ Сработает ли алерт, если бэкап умышленно «сломать»? Триггер проверен в бою?
- ☐ Есть протокол последнего тестового восстановления с фактически измеренным RTO?
- ☐ Хватит ли места под ретенцию 7/4/6/2 с учётом годового прироста данных +20–30%?
- ☐ Распечатана ли процедура восстановления и доступна без работающей ИТ-системы?

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 и классификация данных по RPO/RTO, расчёт объёмов, смета NAS + S3 под профиль офиса
- Развёртывание всех трёх уровней (rsync, Borg/Restic, S3 с Object Lock) за 2 рабочих дня
- Подключение к нашему Zabbix: триггеры возраста, размера и целостности, алерты в Telegram
- Ежемесячные тестовые восстановления с протоколом и печатной процедурой восстановления
- Сопровождение по абонентке: ротация ключей, замена дисков, ежеквартальные DR-прогоны

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** BorgBackup Docs: borg serve --append-only, encryption repokey-blake2 (borgbackup.readthedocs.io — 1.4.5)
- 02** Restic Docs: Preparing repositories (S3), check --read-data-subset, forget... (restic.readthedocs.io — 0.19.1)
- 03** rsync(1): --link-dest, -aHAX, --numeric-ids (download.samba.org — 3.4.4)
- 04** systemd.timer / systemd.exec: OnCalendar, Nice, IOScheduling* (freedesktop.org — v258)
- 05** Yandex Object Storage: Object Lock (governance/compliance) (yandex.cloud — 2026)
- 06** Zabbix Docs: zabbix_agent2, UserParameters (zabbix.com — 7.0 LTS)
- 07** Шаблон ITfresh: backup-shares.sh, ротация daily.N, маркеры .success (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

