



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Duplicati в офисе: бэкап в облако без боли

Как мы ставим Duplicati 2.3 на S3: служба, шифрование, retention и репетиция restore

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Файловый сервер и база 1С живут на одном RAID, а «синхронизация в облако» послушно повторяет за шифровальщиком удаление и подмену файлов. Мы решаем задачу независимой копии: данные уезжают в объектное хранилище зашифрованными на стороне клиента, с версиями и заданной глубиной, и их нельзя стереть с заражённого сервера. Иначе отказ диска или шифровальщик — остановка компании.

Почему это важно бизнесу

- Простой офиса на 20-50 мест — это вставшие отгрузки и оплаты: каждый день без базы 1С оплачивается зарплатой и потерянной выручкой
- Утрата учётных регистров и первички — риск доначислений и штрафов при налоговой проверке, документы придётся собирать заново у контрагентов
- Диск-синхронизация не заменяет бэкап: шифровальщик догоняет копию за часы, а глубины версий там либо нет, либо она измеряется днями
- Бэкап без проверенного восстановления — расход без результата: узнавать о нерабочей копии в час аварии обходится дороже всего проекта
- Один облачный провайдер — одна точка отказа: авария или блокировка аккаунта на сутки останавливает восстановление целиком



Ключевые параметры реализации

2.3.0.4

Актуальный stable (07.2026) на .NET 10 — в прод берём только stable, canary держим на стенде

релиз-ноты Duplicati 2.3.0.x

1 MB

--blocksize для наборов 100 ГБ-1 ТБ вместо дефолтных 100 KB: компакнее локальная SQLite-БД

наш стандарт поверх доков Duplicati (choosing sizes)

50 MB

--dblock-size оставляем дефолтным: размер тома .dblock, меняем только под лимиты бэкенда

дефолт --dblock-size, доки Duplicati 2.3.x

12M:1M

хвост нашей retention-policy 1W:1D,4W:1W,12M:1M — год месячных срезов поверх недельных

Smart retention, доки Duplicati 2.3.x

8200/tcp

порт веб-UI; --webservice-interface=loopback (значение по умолчанию), наружу не публикуем

дефолт Duplicati 2.3.x, наш стандарт

5

--backup-test-samples: поднимаем с дефолтной 1 до 5 наборов томов, проверяемых после прогона

дефолт 1 по докам, наш стандарт 5



Установка службой и изоляция веб-интерфейса

Что настраиваем

Windows Server 2019/2022 и Debian 12 / Ubuntu 24.04 в роли файлового сервера и сервера 1C

Как мы это делаем

- 1 Windows: службу ставит MSI либо регистрируем сами —
Duplicati.WindowsService.exe INSTALL
--webinterface-interface=loopback --server-datafolder=<путь>,
автозапуск с ОС
- 2 Linux: пакет из релизов проекта, systemctl enable --now
duplicati.service (контекст root, настройки в /root/.config/Duplicati); к
UI — ssh -L 8200:localhost:8200
- 3 Пароль веб-UI отключить нельзя: свой задаём
--webinterface-password, поля настроек шифруем
--settings-encryption-key или переменной
SETTINGS_ENCRYPTION_KEY
- 4 Служба Windows идёт от LocalSystem — у неё уже есть
SeBackupPrivilege для VSS; переводим на отдельную учётку только
вместе с правами на снапшоты и исходные каталоги
- 5 --server-datafolder и локальные SQLite-БД заданий (--dbpath)
выносим на отдельный том, чтобы они не конкурировали за место
с общими папками

РЕЗУЛЬТАТ

Бэкап перестаёт зависеть от того, залогинен ли админ: служба стартует вместе с ОС, задания идут ночью, веб-UI слушает только loopback и не попадает в сканы периметра. Служебные данные и базы заданий лежат отдельно от продуктивных томов.

КЛЮЧЕВОЙ НЮАНС

Если чувствительные поля Duplicati-server.sqlite уже зашифрованы, сервер откажется стартовать без ключа, а --require-db-encryption-key запрещает старт без него: ключ кладём в хранилище паролей заранее.



Задание в S3 с шифрованием на стороне клиента

Что настраиваем

Общие папки файлового сервера и каталог выгрузок `.dt / pg_dump` в S3-совместимое объектное хранилище

Как мы это делаем

- 1 Destination «S3 Compatible»: адрес хранилища в `s3-servername`, ключи сервисного аккаунта и обязательно `--use-ssl=true` — по умолчанию S3-канал не шифруется
- 2 Не-AWS-хранилище: либо `--s3-disable-chunk-encoding`, либо клиент `--s3-client=minio` (не оба сразу), регион бакета — `--s3-location-constraint`
- 3 Шифрование: модуль `aes` (`--encryption-module=aes`, AES-256), длинную парольную фразу генерируем и заводим в менеджер паролей; `--no-encryption` регламентом запрещён
- 4 Размеры: `--blocksize=1MB` для наборов от 100 ГБ вместо дефолтных 100 KB (задаётся только при создании), `--dblock-size` оставляем 50 MB под лимиты бэкенда
- 5 Канал и устойчивость: `--throttle-upload` под ширину апlinkа, дефолтные `--number-of-retries=5` и `--retry-delay=10s` повышаем на нестабильных линках

РЕЗУЛЬТАТ

Получаем зашифрованный на стороне клиента набор с дедупликацией на уровне блоков: изменённый мегабайт в гигабайтном файле уезжает несколькими блоками, а не целым файлом. Провайдер хранит только `.dblock/.dindex/.dlist` и не видит содержимого.

КЛЮЧЕВОЙ НЮАНС

`blocksize` фиксируется в момент создания набора и потом не меняется. Для сотен гигабайт сразу ставим 1 MB: со 100 KB локальная SQLite-БД разрастается и задание деградирует, лечится только пересозданием набора.



Контроль результата и репетиция восстановления

Что настраиваем

Все задания на всех узлах клиента плюс кварталный регламент тестового восстановления

Как мы это делаем

- 1 `--run-script-after`: скрипт читает `DUPLICATI_PARSED_RESULT` (Success/Warning/Error/Fatal) и `DUPLICATI_OPERATIONNAME`, шлёт итог в Telegram и почту дежурного
- 2 Держим скрипт коротким: `--run-script-timeout` по умолчанию 60s, тяжёлый разбор `DUPLICATI_RESULTFILE` выносим в отдельный обработчик вне таймаута
- 3 Целостность хранилища: `--backup-test-samples=5` (или `--backup-test-percentage`) после каждого прогона, раз в месяц полная `uplicati-cli test <url> all`
- 4 Раз в квартал восстанавливаем случайный файл недельной давности, раз в полгода — дампы 1C на стенд; при потере сервера — Direct restore from backup files
- 5 При рассинхроне БД и хранилища: `list-broken-files` → `affected` → `repair`, и только осознанно `purge-broken-files`, зафиксировав список теряемых версий

РЕЗУЛЬТАТ

Провал задания виден в тот же час, а не через полгода. Выборочная проверка томов после каждого прогона ловит битые и пропавшие файлы в хранилище до того, как они понадобятся в аварии, а кварталные репетиции подтверждают реальный RTO.

КЛЮЧЕВОЙ НЮАНС

Восстановление на новом железе держится на трёх вещах: URL хранилища, ключи доступа и парольная фраза. Проверяем, что этот набор хранится вне восстанавливаемого контура, иначе авария забирает и его.



Подводные камни

✗ **Потеря парольной фразы = потеря архива**

Ключ шифрования есть только у вас, провайдер не поможет. Держим фразу в менеджере паролей с 2FA плюс офлайн-копию, ротацию пишем в паспорт клиента.

✗ **blocksize задним числом не поменять**

Параметр фиксируется при создании набора. Взяли дефолтные 100 KB под терабайтный сервер — получите распухшую SQLite-БД и только пересоздание набора.

✗ **Lifecycle на бакете съедает историю**

Правило «удалять объекты старше N дней» уносит старые .dblock, и глубокие версии не восстановить. Ротацию отдаём только retention-policy Duplicati.

✗ **VSS включается не при любом запуске**

Дефолт off; с 2.1.1.0 VSS включается сам, только при SeBackupPrivilege у процесса. Ставим --snapshot-policy=required и расширяем квоту теневых копий.

✗ **Веб-интерфейс 8200 смотрит наружу**

Панель с ключами S3 в интернете — подарок сканерам. Оставляем loopback, не ставим --webservice-interface=any, входим SSH-туннелем или через VPN.

✗ **Мониторим запуск, а не результат**

Служба жива, а задание падает с истёкшим AccessKey. Разбираем DUPLICATI__PARSED_RESULT в run-script-after и следим за давностью последнего Success.

✗ **Локальная БД растёт, compact буксует**

Порог сжатия --threshold=25 по умолчанию; при массовых удалениях нужен vacuum или --auto-vacuum=true, иначе SQLite пухнет и прогон идёт всё дольше.

✗ **Первый полный бэкап рвёт канал**

Терабайты через офисный аплинк уезжают сутками. Ставим --throttle-upload, стартуем на локальный диск, затем переключаем задание на S3 — уедет дельта.



Как правильно

МИНИМУМ

- Duplicati 2.3.x stable как служба с автозапуском, веб-UI только на loopback
- Шифрование модулем aes (AES-256), парольная фраза в менеджере паролей
- Ежедневное задание с retention-policy и уведомлением по итогу прогона
- Проверка целостности после каждого прогона (--backup-test-samples=5)

НОРМАЛЬНО

- Два независимых хранилища в разных локациях, lifecycle на бакетах отключён
- --blocksize подобран под объём набора, --dblock-size под лимиты бэкенда
- run-script-after с разбором DUPLICATI__PARSED_RESULT в Telegram и почту
- Квартальный тестовый restore случайного файла с протоколом и замером времени

ХОРОШО

- Zabbix: давность последнего Success, размер набора, число warning в логе
- Шифрование настроек сервера (--settings-encryption-key) на всех узлах
- Полугодовое восстановление дампа 1С на изолированный тестовый стенд
- Отдельный S3-аккаунт бэкапа без прав на удаление, versioning на бакете



Чек-лист самопроверки

- | | |
|---|---|
| ☐ Duplicati запущен службой/systemd-юнитом, а не в сессии пользователя, и стартует вместе с ОС? | ☐ После каждого прогона выполняется проверка томов (--backup-test-samples выше дефолтной 1)? |
| ☐ Веб-UI слушает loopback, пароль --webservice-password задан, доступ только туннелем или через VPN? | ☐ Уведомление приходит по результату задания (PARSED_RESULT), а не только по факту запуска? |
| ☐ Парольная фраза AES-256 сохранена в двух независимых местах и проверена на реальном restore? | ☐ Есть вторая копия в другом хранилище или локально — правило 3-2-1 закрыто фактически? |
| ☐ На бакете нет lifecycle-правил, способных удалить старые .dblock раньше срока хранения? | ☐ Тестовое восстановление файла проводилось за последние 3 месяца и оформлено протоколом? |
| ☐ Задана retention-policy, и глубина хранения совпадает с требованиями бизнеса и бухгалтерии? | ☐ Для открытых файлов и баз задан --snapshot-policy=required и хватает квоты теневых копий? |

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 данных и схема «что бэкапим — куда — как часто — на какую глубину хранения»
- Развёртывание Duplicati 2.3.x службой на Windows и Linux, задания и шифрование AES-256
- Подключение S3-хранилищ, вторая площадка у другого провайдера, отключение lifecycle
- Мониторинг: run-script-after в Telegram и шаблон Zabbix по давности и размеру набора
- Регламент репетиций: квартальное восстановление файла и полугодовое — дампа 1С

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** Duplicati Docs — Advanced options (blocksize, dblock-size, threshold) (docs.duplicati.com — 2.3.x)
- 02** Duplicati Docs — Retention settings (retention-policy, smart retention) (docs.duplicati.com — 2.3.x)
- 03** Duplicati Docs — Scripts: run-script-before/after и переменные среды (docs.duplicati.com — 2.3.x)
- 04** Duplicati Docs — S3-compatible destination (s3-servername, s3-client) (docs.duplicati.com — 2.3.x)
- 05** Duplicati Docs — Windows Service и обработка заблокированных файлов (docs.duplicati.com — 2.3.x)
- 06** Duplicati Docs — The server database и шифрование настроек (docs.duplicati.com — 2.3.x)
- 07** Release notes v2.3.0.x_stable проекта Duplicati (.NET 10) (github.com/duplicati — 2026)
- 08** Шаблон ITfresh: регламент проверки restore и контроль в Zabbix (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

