ASTRAVERGE / STORAGE

Администрирование

AstraVerge Storage 1.4.0

Практическое руководство · 9 ноября 2026

Объектный доступ поверх вашей инфраструктуры.
Files · Pebble · PostgreSQL · AWS S3 / MinIO

Команды и имена файлов сохраняют техническое имя av-storage. Параметры и границы применимости относятся к версии 1.4.0.

Содержание

Устройство и владение данными 3

Что означает успешная запись 3

Отсутствие ответа и повторы 4

Процесс сервера и providers 4

WAL и recovery 4

Мониторинг и ежедневная диагностика 5

Backup и восстановление 5

Обновление, конфигурация и миграция 6

Пределы гарантий 6

Версия контракта: AstraVerge Storage 1.4.0. Установку см. здесь, все параметры - в CONFIGURATION. Это руководство определяет порядок работы с данными и действия при сбоях.

Устройство и владение данными

Listeners REST/gRPC/client IPC проверяют authentication/ACL, затем передают объектную операцию Router. Registry связывает logical storage с backend. Files/Pebble открываются внутри сервера; PostgreSQL/S3 запускаются отдельным child process **на каждый logical storage**, с собственным пулом IPC connections. Данные удалённого backend не находятся в локальном каталоге provider.

Data-root принадлежит одному writer, сервер удерживает lock. Backend files/Pebble имеют собственную защиту владения. Не обходите lock удалением sidecars и не открывайте рабочий каталог сторонним writer. Изменения данных напрямую в backend обходят WAL/overlay/ACL AstraVerge Storage и требуют отдельной координации.

Что означает успешная запись

Режим	Условие успешного ответа
Обычный PUT/DELETE, wal_enabled=true	Полная запись принята WAL; provider application асинхронно
Обычный PUT/DELETE, wal_enabled=false	Backend вернул успех; серверного recovery для новой записи нет
PUT_BATCH ram	Копия принята в ограниченную volatile RAM queue
PUT_BATCH flush	Каждая запись завершила обычный путь PUT
PUT_BATCH sync	Каждая принятая запись прошла explicit WAL Sync либо допустимый files backend Sync

По умолчанию WAL включён, `batch_ack_mode=flush`, `fsync_every_records=0`. Flush означает передачу записи файловому API/OS cache, **не обязательный fsync**. При N=1 синхронизация инициируется на каждой записи; при N>1 первые ACK группы могут предшествовать Sync. Sync batch форсирует границу синхронизации и требует N>0. Проверить эффективный контракт: `av-storage -config av-storage.conf -inspect-config`; при monitoring он также доступен в status.

Успешный PUT не означает, что объект уже находится в PostgreSQL/S3/files backend. GET может обслужить pending PUT прямо из WAL; pending DELETE возвращает NotFound. Поэтому успешное контрольное чтение подтверждает логическую видимость через сервер, но не завершение backend application. RAM batch видимость обеспечивается ordering barriers перед последующими операциями; при crash ещё не сброшенные RAM records могут потеряться, несмотря на полученный ACK.

Batch не является транзакцией: при ошибке возможен принятый префикс. Snapshot consistency, exactly-once и транзакции между объектами/storages не предоставляются. Для GET_BATCH успех - только терминальный END/OK с полным ожидаемым числом records и штатным EOF; начало ответа не доказывает завершение.

Отсутствие ответа и повторы

Timeout/disconnect - неопределённый результат: запись могла уже попасть в WAL или backend. Отмена клиента не откатывает принятые изменения. Повтор PUT с теми же bytes по тому же адресу возвращает то же итоговое значение только при отсутствии конфликтующих изменений. Если другой writer успел записать новую версию, слепой retry может её перезаписать. Выделяйте стабильные ключи операций/версий и координируйте writers в приложении; CAS и idempotency key API нет. Повтор DELETE также требует учёта параллельного создания объекта. S3 versioning может создавать дополнительные версии/delete markers при повторном применении.

Используйте ограниченный retry с backoff и jitter для transient unavailable/transport error; не повторяйте бесконечно invalid address, неверные credentials или ACL denial. Для batch считайте весь неопределённый запрос потенциально частично принятым. Проверка GET помогает установить текущее значение, но не доказывает отсутствие более раннего применения или конкурентных изменений.

Процесс сервера и providers

Startup: загрузка config → security/TLS validation → lock data-root → discovery/open storages → WAL recovery → listeners → READY. Ошибка одного enabled backend может сорвать весь startup. READY означает запуск интерфейсов, а не обязательное завершение всего recovered WAL. `ping` проверяет доступность client IPC, не все backend и не прикладные ACL. Проверяйте отдельный storage контролируемым CRUD и наблюдением pending WAL.

При падении child supervisor закрывает соединения старой generation и примерно через секунду пытается перезапустить executable; неудачный start повторяется. Пул пересоздаёт connections по мере обращений. Этот механизм не добавляет HA сервера и не переключает хранилище на другой host. ProcessStore сам не повторяет consumed request; replay принятых изменений выполняет Router/WAL. Чтения во время перезапуска могут получить unavailable.

Остановка: прекратите новую нагрузку, завершите клиентские streams/readers, отправьте Ctrl+C/SIGTERM и дождитесь выхода процесса. Сервер прекращает admission, сбрасывает принятые RAM batches и закрывает Router/backend. Уже принятый WAL может остаться для следующего запуска. Shutdown не имеет универсального deadline для зависшего backend/удерживаемых WAL readers. На Windows не заменяйте Ctrl+C обычным принудительным kill в ежедневном регламенте. Для restart дождитесь полного освобождения lock/IPC и запускайте тот же комплект с теми же путями.

WAL и recovery

Адрес стабильно хешируется в одну FIFO lane; операции этой lane применяются последовательно. Worker повторяет ошибку backend примерно раз в секунду, не пропуская head. Недоступный storage способен задержать другие адреса той же lane. Другие lanes продолжают работу. При включённом WAL новые изменения зарегистрированного storage могут приниматься без предварительного health-check provider; это требует контроля диска и backlog.

При startup восстанавливаются полные неприменённые records, overlay и очередь применения. Неполный хвост WAL обрезается; bad magic/некорректная позиция являются ошибками, не приглашением удалить журнал. Применение может повториться, если backend завершил операцию, а checkpoint ещё не опубликован, либо консервативный reset допустил replay. DELETE NotFound при replay считается успехом.

Сохраняйте logical names, backend mapping, WAL directory и lanes. При pending нельзя менять lanes; отключение WAL для storage с recovered pending отвергается. Не удаляйте `.wal`, `.pos` или layout metadata

вручную. Ошибка WAL Ack логируется и может остановить worker lane; supervisor child её не исправляет. Сначала сохраните согласованный stopped snapshot, устраните диск/права, затем выполните контролируемый restart. При повреждении не редактируйте offsets наугад - используйте проверенный backup и инженерный разбор копии.

Физический размер WAL не равен pending bytes. GET reader закрепляет lane от truncate/reuse; даже применённые данные могут удерживаться до закрытия reader. Долгий GET способен увеличить диск и задержать shutdown.

Мониторинг и ежедневная диагностика

Включите `[monitoring] enabled=true`, `listen="127.0.0.1:9108"`; при необходимости задайте отдельный token. `av-storage-admin -command status -monitoring http://127.0.0.1:9108` читает `/status`; token берётся из `AV_STORAGE_MONITORING_TOKEN`. `/metrics` - Prometheus. Это отдельный ops endpoint, не storage ACL. Для удалённого сбора используйте локальный агент/защищённый proxy; listener принимает только loopback IP.

Наблюдайте pending records/bytes и возраст старейшего record, RAM queue, failed/busy storage, ошибки и latency слоёв, свободное место WAL/backend/temp и жизнь процессов. Pending bytes - логический неприменённый объём, не физический размер файлов; recovered age отсчитывается от запуска. Busy exporter не ждёт занятый backend lock, поэтому storage queue при busy не является актуальным снимком. Go memory - не RSS/page cache. Latency histogram слоёв перекрываются, их нельзя складывать как независимые этапы.

Примеры Prometheus, alerts и Grafana находятся в examples/monitoring. Порог backlog подбирайте по нагрузке; продукт не задаёт SLA. Логи stdout/stderr собирайте внешним supervisor с ротацией. Встроенных log_level/log_file нет. Audit имеет отдельную семантику.

Backup и восстановление

Штатный admin поддерживает **offline physical backup локальных files/Pebble**. Остановите server и всех независимых владельцев backend. Выходной архив должен находиться вне data-root и не существовать. External/remote provider, symlink/special files и внешние WAL/backend paths отклоняются; не обходите эту проверку.

```
.\av-storage-admin.exe -command backup -config C:\AV\av-storage.conf -data C:\AVData -archive
C:\Backups\av-001.tar
.\av-storage-admin.exe -command restore -archive C:\Backups\av-001.tar -data C:\AVRestored -max-bytes
1099511627776
.\av-storage.exe -config C:\AVRestored\recovered-server.conf -data C:\AVRestored
```

Target restore должен отсутствовать. Restore проверяет размеры/SHA-256, traversal, duplicates и links, извлекает в отдельную область и публикует результат после проверки. При ошибке backup может остаться неполный архив: не считайте наличие файла успехом, проверяйте exit code. На Unix используйте аналогичные пути и executable без `.exe`.

В архив входят data-root с backend/WAL и основной config; lock sidecars исключены. Основной config может содержать DSN/monitoring secret - защищайте архив. Отдельные security file, cert/key не становятся автоматически полной резервной копией инфраструктуры: сохраните их отдельным защищённым регламентом и после restore проверьте paths. Не отправляйте restored config в production, пока не проверены destinations и credentials.

Для PostgreSQL используйте согласованный backup СУБД, для S3 - его versioning/backup/replication по политике владельца backend. AstraVerge Storage не предоставляет согласованного online distributed snapshot нескольких backend и своего WAL. План восстановления удалённого backend должен согласовывать replay: старый WAL может повторно заменить/удалить объекты. Проверяйте disaster recovery на отдельном окружении, включая фактические bytes, listing и ACL.

Обновление, конфигурация и миграция

До обновления сохраните предыдущие binaries/config, сделайте проверенный backup и контрольный restore. Остановите нагрузку, дождитесь применения WAL по monitoring (при необходимости), остановите сервер, замените согласованный комплект сервер/CLI/providers. Не смешивайте неизвестные версии протокола. Проверьте `-version`, `-inspect-config`, startup, CRUD, listing, monitoring и security на тестовом namespace. 1.4.0 сохраняет IPC v4/backend/WAL formats, но добавляет defaults 1 GiB на REST/gRPC и строгую проверку полей listeners: учтите большие существующие объекты.

Конфигурация применяется только restart. Для отката остановите новый процесс и верните старый комплект с совместимым config; при изменениях данных/форматов используйте заранее проверенный restore-план, а не обещание универсального downgrade. Исторический provider-addressed WAL до 0.9 не сопоставляется автоматически с logical names.

Смена provider не переносит объекты. Даже неизменное logical name с другим bucket/prefix/path меняет destination recovery. План переноса: остановить writers → завершить pending application → сохранить backup → выгрузить объекты через поддерживаемый клиент с manifest key/size/hash → записать в отдельный target storage → проверить все bytes/keys → переключить приложение/config → сохранить исходный backend до приёмки. Это ручной проект миграции; встроенного migration command нет. Для больших namespace учтите полную материализацию LIST_KEYS. Не выполняйте одновременное изменение маршрута и replay старых pending records.

Пределы гарантий

Crash tests завершают процесс без Close и проверяют replay, в том числе состояния до/после backend application/checkpoint. Это не испытания отключением питания. Files content fsync не доказывает сохранность directory metadata при аппаратном отказе; универсального directory-fsync контракта нет. Native WAL Pebble и remote PostgreSQL/S3 имеют собственные политики; AstraVerge Storage не управляет remote fsync. При native Pebble WAL off сервер выполняет checkpoint перед признанием применения.

Нет встроенного cluster membership, replication, consensus, HA, host failover, автоматической миграции, неограниченного масштабирования или публичного SLA. Отказ локального WAL-диска требует инфраструктурной защиты и backup. Практический справочник.