ASTRAVERGE / STORAGE

Установка и первый запуск

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

Платформы и ресурсы 3

Получение и проверка поставки 3

Что находится в архиве 4

Первый конфигурационный файл 4

Запуск и остановка 5

Первое подключение и CRUD 5

Подключение удалённых backend и служб ОС 6

AstraVerge Storage предоставляет приложениям операции над бинарными объектами по адресу `(storage, namespace, key)`. Здесь описан самостоятельный запуск поставки, без доступа к исходникам. Для production после первого локального запуска выполните настройку безопасности и подготовьте эксплуатационный регламент.

Платформы и ресурсы

В репозитории есть архивы **Windows/amd64, Linux/amd64 и FreeBSD/amd64**. Поставки собраны с CGO_ENABLED=0. Запуск и smoke текущего релиза подтверждены на Windows; для Linux/FreeBSD подтверждены кросс-сборка, формат исполняемых файлов и права в архиве. Нативные испытания этих двух ОС в релизном отчёте отсутствуют. Минимальные версии ОС и аппаратные минимумы не установлены испытаниями. ARM-сборки в поставке отсутствуют.

Нужны x86-64, права записи в каталоги данных/WAL/runtime и свободные выбранные порты. Go для готовых программ не требуется. Для сборки из исходников требуется Go 1.27; релиз собран Go 1.27.3. Размер RAM и диска выбирайте по нагрузке: Pebble и PostgreSQL материализуют один объект в памяти; учитывайте параллельные операции, кеш backend и WAL при недоступности хранилища. Лимиты очередей не являются пределом RSS. Диск должен вмещать backend, неприменённый WAL, временные загрузки и резервную копию вне data-root.

Получение и проверка поставки

Выберите архив своей ОС из `releases/1.4.0`. Сверьте SHA-256 с переданным по доверенному каналу manifest/SHA256SUMS.txt. Хеш проверяет целостность, но не заменяет подпись издателя.

Windows, PowerShell (из каталога с архивом):

```
Get-FileHash .\av-storage-1.4.0-windows-amd64.zip -Algorithm SHA256
Expand-Archive .\av-storage-1.4.0-windows-amd64.zip -DestinationPath C:\AstraVerge\Storage-1.4.0
Set-Location C:\AstraVerge\Storage-1.4.0
```

Linux:

```
sha256sum av-storage-1.4.0-linux-amd64.tar.gz
mkdir av-storage-1.4.0-linux-amd64
tar -xzf av-storage-1.4.0-linux-amd64.tar.gz -C av-storage-1.4.0-linux-amd64
cd av-storage-1.4.0-linux-amd64
```

FreeBSD:

```
sha256 av-storage-1.4.0-freebsd-amd64.tar.gz
mkdir av-storage-1.4.0-freebsd-amd64
tar -xzf av-storage-1.4.0-freebsd-amd64.tar.gz -C av-storage-1.4.0-freebsd-amd64
cd av-storage-1.4.0-freebsd-amd64
```

Штатные архивы содержат файлы без внешнего каталога версии, поэтому распаковывайте их в заранее выбранный отдельный каталог. Дальнейшие команды выполняются рядом с `av-storage` и `av-storage.conf.example`.

Что находится в архиве

Файл	Назначение
av-storage[.exe]	Сервер, Router, WAL, встроенные files/Pebble
av-storage-cli[.exe]	Локальный клиент IPC; генерация токенов
av-storage-grpc-cli[.exe]	Клиент gRPC
av-storage-admin[.exe]	Status и offline backup/restore
av-storage-stream-example[.exe]	Пример пакетной записи и потокового чтения
av-storage-bench / av-storage-crash	Инструменты испытаний; не запускать на рабочих данных
providers/av-storage-provider-postgres	Внешний PostgreSQL provider
providers/av-storage-provider-s3	Внешний AWS S3/MinIO provider
providers/av-storage-provider-pebble	Совместимая утилита; штатный сервер использует встроенный Pebble
*.conf.example, docs/, examples/	Шаблоны и справочные материалы
release.json, SHA256SUMS.txt	Происхождение и целостность файлов

Расширение `.exe` используется на Windows. Полный состав зафиксирован в материалах релиза.

Первый конфигурационный файл

Создайте `av-storage.conf` рядом с сервером со следующим содержимым. Это самостоятельный минимальный пример, а не дополнение к шаблону со включённым PostgreSQL.

```
[storages]
enabled = "files, local"
[storage.files]
provider = "files"
[storage.local]
provider = "pebble"
[rest]
enabled = true
listen = "127.0.0.1:8080"
[grpc]
enabled = true
listen = "127.0.0.1:9090"
[monitoring]
enabled = true
listen = "127.0.0.1:9108"
[wal]
lanes = 16
path = "wal"
```

Security здесь выключена по умолчанию. Пример рассчитан на доверенную локальную проверку. Не открывайте его сетевые listeners наружу. Запускайте из выделенного каталога от отдельной учётной записи с доступом только к нужным данным. Сервер создаёт data-root, `storages` и WAL самостоятельно. Для Unix

IPC создаётся `run/`, сокет имеет mode 0600. Windows named pipe использует платформенный default security descriptor: отдельный настраиваемый ACL трубы в конфигурации не предусмотрен.

Запуск и остановка

Windows:

```
.\av-storage.exe -version
.\av-storage.exe -config .\av-storage.conf -inspect-config
.\av-storage.exe -config .\av-storage.conf -data .\data -providers .\providers
```

Linux/FreeBSD:

```
./av-storage -version
./av-storage -config ./av-storage.conf -inspect-config
./av-storage -config ./av-storage.conf -data ./data -providers ./providers
```

Дождитесь `READY`, проверьте имена storages и адреса в выводе. `-inspect-config` проверяет runtime-настройки и показывает границы ACK; не проверяет подключение backend, сертификаты и весь security file. Отсутствующий основной config даёт defaults, поэтому всегда проверяйте список Storages: неверный путь не обязательно завершит запуск ошибкой.

Для остановки foreground-сервера нажмите Ctrl+C; Unix также обрабатывает SIGTERM. Дождитесь выхода процесса, прежде чем запускать новый. Завершение не означает, что весь WAL уже применён: остаток сохраняется для recovery. Зависший backend или удерживаемый reader могут задержать выход. Не удаляйте WAL для ускорения остановки.

Первое подключение и CRUD

Откройте вторую консоль в том же каталоге. Windows:

```
.\av-storage-cli.exe ping
.\av-storage-cli.exe put files demo hello "HELLO"
.\av-storage-cli.exe get files demo hello
.\av-storage-cli.exe list-keys files demo
.\av-storage-cli.exe delete files demo hello
.\av-storage-cli.exe list-keys files demo
.\av-storage-admin.exe -command status -monitoring http://127.0.0.1:9108
```

Ожидается `OK`, затем `HELLO`, список с `hello`, после удаления - пустой список `[]`. PUT/DELETE при успехе не печатают сообщение; проверяйте exit code. Замените `files` на `local`, чтобы проверить Pebble. На Unix команды те же с `./av-storage-cli` и `./av-storage-admin`, без `.exe`. Выполняйте из того же cwd: Unix endpoint по умолчанию относительный.

REST, PowerShell:

```
curl.exe --fail-with-body -X PUT --data-binary "HELLO REST" http://127.0.0.1:8080/v1/files/demo/hello
curl.exe --fail-with-body http://127.0.0.1:8080/v1/files/demo/hello
curl.exe --fail-with-body -X DELETE http://127.0.0.1:8080/v1/files/demo/hello
.\av-storage-grpc-cli.exe -endpoint 127.0.0.1:9090 put files demo hello "HELLO GRPC"
.\av-storage-grpc-cli.exe -endpoint 127.0.0.1:9090 get files demo hello
```

На Unix используйте `curl` и `./av-storage-grpc-cli`. LIST_KEYS доступен только через client IPC. GET после DELETE должен сообщить NotFound, REST - 404. Матрица API объясняет транспортные различия.

Подключение удалённых backend и служб ОС

Для PostgreSQL/S3 сначала подготовьте backend и выполните явный `-init` из руководства настройки. Обычный запуск не создаёт PostgreSQL schema/table или S3 bucket.

В поставке нет Windows service installer, готового systemd unit или FreeBSD rc.d-скрипта. `sc.exe create` сам по себе не превращает foreground-программу в Windows service. Для постоянной эксплуатации администратор должен отдельно интегрировать процесс с принятым supervisor, задав cwd, явные пути, окружение, учётную запись, сбор stderr и корректный сигнал остановки. Настройка такой интеграции требует проверки на целевой ОС; она не входит в подтверждённый установщик AstraVerge Storage.