

Abkus Backup - help

The backup catalog and folder viewer of the Abkus fleet. It shows what exists, exactly where, when it was made, whether it is encrypted and whether a restore was proven - and it says when something is missing, stale or unproven.

Index

- How it works
- The backup catalog
- Folder contents - source and destination
- Coverage - is there a recent copy?
- Restore proof - prove, don't promise
- Off-site package
- Retention - is there enough history?
- The single verdict
- Setup
- Plans, cancellation and refunds
- Support

How it works

From the backup job to the one verdict, in three steps.

1. The daily backup job writes the copies and a manifest on the server. The catalog reads that manifest - it never connects to your database.
2. Each host publishes a folder listing. The app compares the source with the destination and counts only real backup copies.
3. Open the dashboard: one verdict lists the gaps (missing, stale, plaintext, off-site, destination, proof), with the check that found each one.

The backup catalog

Every backup artifact with the facts you asked for.

For each copy the catalog shows: what it is (database dump, archive, git bundle), the exact host and folder, when it was made, its size, whether it is encrypted and its hash.

It is read-only: the catalog opens, reads the leading bytes and hashes - it never writes, moves or deletes a backup.

Hashing is bounded so a large folder does not hang the page. Set `ABKUS_BACKUP_HASH` to "all", "none" or a byte budget (default 256 MiB); above the budget the artifact is listed without a hash.

Folder contents - source and destination

See the files and folders each host really holds, on both sides of the copy.

A listing is capped at 2000 entries per root on the host. A capped listing is marked Truncated and is never treated as complete.

Inventories are snapshots. A listing older than `ABKUS_BACKUP_INVENTORY_MAX_AGE_HOURS` (default 26), or dated in the future, is marked Stale and never treated as current.

1. On each VPS/computer, run the collector: `node scripts/collect-inventory.ts --host <name> --role source|destination --root <folder> --out <inventories>/<host>.json`
2. Point `ABKUS_BACKUP_INVENTORIES_DIR` (on the app host) at the folder that holds those JSON files.
3. On the app host itself, set `ABKUS_BACKUP_INVENTORY_ROOTS` (colon-separated folders) and

ABKUS_BACKUP_INVENTORY_ROLE.

4. Open Browse: each host is listed with its role, age and files/folders (name, path, kind, size, modified).

5. The Source vs destination section shows which backup copies the destination does not hold and any size mismatch. Metadata files (the manifest, inventories) are not counted as copies.

Coverage - is there a recent copy?

Per source: fresh, stale or missing.

Set ABKUS_BACKUP_EXPECT to lines id:pattern:maxAgeHours (for example abkus_hub:abkus_hub-*.sql.gz:26). The catalog then reports, per source, the newest copy and whether it is fresh, stale or missing.

A source with no copy is critical; only a stale copy is attention.

Restore proof - prove, don't promise

A copy is only trustworthy after a restore was actually tested.

Point ABKUS_BACKUP_PROOF at the restore drill report and set ABKUS_BACKUP_PROOF_EXPECT to lines source:maxAgeDays.

Each source is then proven, stale, failed or never proven. A failed restore is critical; a missing or stale proof is attention. A source with no proof is never assumed green.

Off-site package

One encrypted package, several destinations - read from the producer.

The daily job writes the encrypted package and ships it to several destinations. When

ABKUS_BACKUP_OFFSITE_STATUS points at offsite-status.json, the dashboard shows the package, its age and every destination.

A destination that failed is attention; a package older than ABKUS_BACKUP_OFFSITE_MAX_AGE_HOURS (or dated in the future) or with no successful destination is critical. The per-artifact "not off-site" flag is only the fallback.

Retention - is there enough history?

Freshness checks recency; retention checks how many copies exist.

Set ABKUS_BACKUP_RETENTION to lines id:pattern:minCopies (for example abkus_hub:abkus_hub-*.sql.gz:7). A source with fewer copies than required is flagged as insufficient retention.

The single verdict

One status for the whole backup picture.

The verdict is ok, attention or critical. Critical: a plaintext copy, a missing copy, a failed restore, an unreadable folder or a stale manifest (the backup job stopped). Attention: a stale copy, an off-site gap, an unconfirmed encryption, insufficient retention or an unproven/stale proof.

The banner also lists which checks actually ran. A check that is not configured is shown as such - it is never counted as passed.

Setup

Where each input comes from.

There is no .env in Abkus: configuration comes from the Hub vault via the environment, never from a committed file.

1. Backup job: writes manifest.json after the upload and copies it off-site. Point ABKUS_BACKUP_MANIFEST at

it.

2. Restore drill: writes the proof report. Point ABKUS_BACKUP_PROOF at it.
3. Policies: ABKUS_BACKUP_EXPECT, ABKUS_BACKUP_PROOF_EXPECT and ABKUS_BACKUP_RETENTION.
4. Freshness of the job: ABKUS_BACKUP_MANIFEST_MAX_AGE_HOURS (default 26).
5. Folder viewer: ABKUS_BACKUP_INVENTORY_ROOTS / _ROLE / _DEPTH and ABKUS_BACKUP_INVENTORIES_DIR.

Plans, cancellation and refunds

Nothing is on sale yet; the guarantee is already defined.

No subscription is on sale yet. When plans launch, every subscription will include a 14-day money-back guarantee and can be cancelled at any time from the account settings.

The full policy is at /refund (Refund Policy).

Support

Stuck or something looks wrong? Ask.

Email support@abkus.com or open the Abkus Support app. Include the host and the problem code shown by the verdict - that is the fastest way to locate the copy.

Support

Email support@abkus.com or open <https://support.abkus.com>.