ALAI Backup Strategy

ALAI Backup Strategy

> ⚠ **STALE-DOC BANNER (2026-08-08, MC #106946):** the Layer 3 block below (SSH to Azure VM `vm-alai-lightrag`, 4 bare volume names) is stale — since the 2026-08-03 migration (#106747) the backup runs entirely local on this Mac against 7 `lr106747-*`-prefixed volumes, no SSH. Corrected, current detail: LightRAG Backup runbook (updated 2026-08-08). Not fully rewriting this page — see MC #106946 evidence for what changed and why.

Status: LIVE — verified against running processes, crontab, launchctl and logs on 2026-07-28. **Owner:** FlowForge (infra) / John (orchestration) **Supersedes for accuracy:** `~/system/architecture/backup-strategy-2026-04-20.md` (that document is a PLAN dated 2026-04-20; several of its proposals were never activated or were later reversed by CEO decision — see §5 "Divergence from the original plan"). This page reflects what is actually running today.

---

1. Architecture — 4 layers, verified live

```
ANVIL (Mac Studio, 100.103.49.98)
│
├── Layer 1 — Git (hourly)
│     tools/hourly-backup.sh — cron 0 * * * *
│     Commits + pushes ~/system, ~/ALAI, ~/.claude, and every repo under ~/projects/*/*
│     to its GitHub/origin remote. Silent-fail-safe: logs push failures, retries next hour.
│
├── Layer 2 — SQLite (continuous + daily + 2x offsite)
│     PRIMARY: litestream replicate daemon (LaunchAgent com.alai.litestream, PID confirmed
│       live 2026-07-28, running since 2026-07-14). Config: ~/system/config/litestream.yml.
│       64 databases, WAL-shipping, sync interval 1s–300s depending on P0/P1/P2 tier.
│       ALL 64 replicas are `type: file` (LOCAL) → ~/backups/litestream/<db>/ (1.1 GB local,
│       verified 2026-07-28). ZERO Azure (`type: abs`) replicas remain — see §5.
│     SECONDARY (local snapshot): tools/db-backup.sh — cron 0 3 * * *, 5-day local retention,
```

```
│       ~/system/backups/databases/
│     OFFSITE #1: tools/rclone-backup.sh — launchd daily 03:00 (com.john.rclone-backup) →
│       Backblaze B2 (b2-alai:alai-studio-backup/system-databases/ + /claude-memory/).
│       Confirmed live run 2026-07-28T03:00Z, OK.
│     OFFSITE #2: daemons/offsite-backup.sh — launchd every 6h (com.john.offsite-backup) →
│       Backblaze B2 (b2-alai:alai-studio-backup/databases/, /config/, /rules/, /specs/,
│       /tools/, /claude-hooks/, /claude-agents/, /claude-skills/, /claude/).
│       Confirmed live run 2026-07-28T03:36Z, 9/9 targets OK, 85s.
│
├── Layer 3 — LightRAG / Neo4j Docker volumes
│     tools/lightrag-backup.sh — launchd weekly Sunday 04:00 (com.alai.lightrag-backup)
│     SSH to Azure VM vm-alai-lightrag (20.240.61.67) → docker compose stop → tar each of
│       4 volumes (lightrag-data, lightrag-kg, lightrag-cache, lightrag-neo4j-data) →
│       docker compose start → SHA-256 manifest.
│     SCP snapshot back to Mac Studio (~/system/backups/lightrag/, local safety net, keep 4).
│     Azure offsite upload → Storage Account `plockfrontstaging` (swedencentral),
│       Container `lightrag-backup`, Cool tier, keep last 8 snapshots (~8 weeks).
│     Full detail: [LightRAG Backup runbook](./lightrag-backup.md)
│
└── Layer 4 — Full ANVIL loss
      No fully-automated one-shot bootstrap is active today. Recovery = re-run Layers 1-3
      restore procedures on a freshly provisioned Mac. See Disaster Recovery Runbook.
```

---

2. Azure storage inventory (tool-verified 2026-07-28)

Account	Region	Purpose	Status

`alaibackups0ebb` (RG `alai-backups-rg`)	—	Originally scoped for SQLite (`system-db-backups`), git bundles (`system-git-bundles`), Bitwarden exports (`bitwarden-exports`) per the April plan	**Scaffolded, not the active path.** LaunchAgent `com.alai.azure-db-backup.plist.disabled` has been disabled since creation (file dated 2026-04-20). Litestream *did* target this account before the 2026-07-13 cost decision moved all 64 DB replicas to local file replicas (see §5). Current default az-cli SP identity gets `AuthorizationFailed` reading this account/RG — could not independently confirm current RBAC state; do not assume access without re-verifying.
`plockfrontstaging` (swedencentral)	Hot	LightRAG/Neo4j weekly offsite (container `lightrag-backup`, Cool tier per-blob)	**Active** — this is a pre-existing Plock storage account being reused for LightRAG backup, not a dedicated ALAI backup account.
Backblaze B2 (`b2-alai:alai-studio-backup`, non-Azure)	—	SQLite snapshots, config/rules/specs/tools, .claude memory/hooks/agents/skills	**Active**, two independent daemons (rclone-backup.sh daily, offsite-backup.sh 6-hourly).

3. Retention policy (as configured, verified)

Layer	Mechanism	Retention
Git	GitHub remote history	Unlimited (git history)
SQLite — litestream local	`~/backups/litestream/<db>/`	24h–168h per DB tier (P0 financial longest)
SQLite — db-backup.sh local	`~/system/backups/databases/`	5 days
SQLite — B2 offsite	rclone sync (mirrors latest snapshot)	Governed by B2 bucket lifecycle (not independently re-verified this session)
LightRAG/Neo4j — local	`~/system/backups/lightrag/`	Last 4 snapshots
LightRAG/Neo4j — Azure (`plockfrontstaging`)	Cool tier blob	Last 8 snapshots (~8 weeks)

The April plan's proposed 30-day-Cool → Archive → 365-day-delete lifecycle on `alaibackups0ebb` applies to the SQLite/git-bundle layer that was never activated in production — treat that lifecycle description as design intent, not live policy.

4. Cost (verified against live config, not the April estimate)

- **SQLite + git layer:** effectively **$0** in direct Azure spend today — litestream runs against local file replicas, git backup is a GitHub push, B2 offsite is Backblaze (separate low-cost provider, not itemized this session).
- **LightRAG/Neo4j layer:** per `lightrag-backup.md`, **~$0.05–0.10/month** on `plockfrontstaging` Cool tier for ~4 GB retained.
- The April plan's **~$2–4/month** estimate was for a dedicated `alaibackups<suffix>` account running the full SQLite+git Azure Blob lifecycle. That path is not the one in production, so that estimate does not describe current spend.

5. Divergence from the original 2026-04-20 plan — why, and the authority for it

CEO decision, verbatim, 2026-07-13 (MC #105462): "*Prebaci lokalno! Nemamo placene korisnika kad to dodje ide backup na azure do tada save money gdje mozemo.*" ("Move it local! We don't have paying customers — when that happens, backup goes to Azure, until then save money wherever we can.")

Context: MC #105462 — litestream's per-sync-tick `ListBlobs` full-prefix enumeration against `alaibackups0ebb` (two hot telemetry DBs, `flywheel`/`mission-control`, at ~113K/~102K blobs each) was driving up to **518 GB/day list-egress (~370 NOK/day)**. After phased mitigation attempts (sync-interval tuning, lifecycle rules) proved insufficient, the CEO's second, later, better-informed decision (received ~10 hours after an earlier, more conservative one — the later decision took precedence per session record) was to migrate **all** litestream replicas to local file targets rather than keep paying Azure list-egress with no paying customers yet. Full incident record: Azure litestream egress saga + backup policy — MC #105462.

Practical effect: the April plan's Layer 2 (SQLite → Azure Blob, dedicated `alaibackups0ebb` account) is stood up in config/scripts but intentionally not the live path. B2 (Backblaze) offsite + local litestream replicas carry that layer today. LightRAG/Neo4j (Layer 3) was unaffected — it is a low-frequency (weekly), low-blob-count workload on a different storage account, so it kept its

original Azure design.

IAM note: the SP configured for the Azure Blob SQLite path (`alai-backup-writer` / appid `1a0b3018-…`, per `~/system/config/litestream.yml` and `azure-backup.env`) is the same appid used as the general-purpose az-cli identity elsewhere in the system. `VAULT_EXPORTER_APPID` in `azure-backup.env` is still `FILL_AFTER_SP_CREATION` — that narrower-scoped SP for Bitwarden exports was never created. Do not assume it exists.

6. Reactivation trigger

Per the CEO decision above, the Azure Blob SQLite backup path (`alaibackups0ebb`) should be reactivated when ALAI has paying customers. Reactivation requires, at minimum: (1) re-verify RBAC on `alai-backups-rg`/`alaibackups0ebb` (current az-cli SP identity failed this check on 2026-07-28), (2) re-enable `com.alai.azure-db-backup.plist` (currently `.disabled`), (3) re-point litestream replicas for at least the P0-financial tier back to `type: abs`, (4) re-run the lifecycle-policy fix from #105490 (an account-wide lifecycle rule without `prefixMatch` archived 35 dormant DBs' L0 blobs during the original incident — must be scoped before reactivation).

Document Owner: Lexicon (Skillforge role) **Last Verified:** 2026-07-28 — process list (`litestream replicate`, live PID), `crontab -l`, `launchctl list`, LaunchAgent plists, live log tails (`offsite-backup.log`, `rclone-backup-20260728.log`), `az account show` / `az storage account show` (AuthorizationFailed, documented as-is), and `~/system/config/litestream.yml` (64/64 DBs confirmed `type: file`).

Revision #2
Created 2026-07-28 06:57:00 UTC by John
Updated 2026-08-08 13:37:33 UTC by John