

LightRAG Backup (Azure-native + local safety net)

LightRAG Backup (Azure-native + local safety net)

“ **Domain note (2026-05-17):** References to `lightrag.basicconsulting.no` in this doc are the legacy hostname. Current live endpoint: `lightrag.alai.no`.

Operational update — 2026-08-19 (MC #107355)

This section supersedes the older Azure-primary statements retained below for migration history. The live source is now the local compose project at `~/system/lightrag-local/docker-compose.local.yml`, backed by the seven `lr106747-*` volumes. The Cloudflare dashboard route for `lightrag.alai.no` targets this machine's `localhost:9621`; `com.john.cloudflare` owns the tunnel process.

The 2026-08-16 weekly backup stopped LightRAG and Neo4j, but LightRAG start raced Neo4j health. `docker compose start lightrag` failed, the script reduced that to a warning, continued through offsite upload, and reported `Done`, leaving the public endpoint at HTTP 502 until 2026-08-19.

The restart contract is now fail-closed:

1. retry start of the existing Neo4j container;
2. wait up to 180 seconds for `lightrag-neo4j` health;
3. retry start of the existing LightRAG container;
4. wait up to 300 seconds for `lightrag` health;
5. exit non-zero with operator-action text if either service fails recovery.

This path uses `docker compose start`; it does **not** recreate containers, delete volumes, or restore data. Public recovery was verified through browser HTTP 200 plus wrapper status, hybrid query, and graph explore. Evidence: `/Users/makinja/system/evidence/107355/REPORT.md`.

Remaining validation: run the next scheduled full backup under observation to prove the new gates end to end. Do not trigger an extra 1.5 GB backup solely as a test because it intentionally causes production downtime.

Owner: FlowForge (infra) **Implemented:** 2026-04-18 (updated for Azure migration 2026-04-18)
Source of Truth: Azure VM `vm-alai-lightrag` (20.240.61.67) **Schedule:** Weekly Sunday 04:00 CEST **Script:** `~/system/tools/lightrag-backup.sh` (SSH-based) **Plist:** `~/Library/LaunchAgents/com.alai.lightrag-backup.plist` **Azure creds:** `~/system/config/azure-lightrag-backup.env` (mode 0600)

What is backed up

4 Docker volumes (+ checksums + README):

Volume	Content	Typical size
<code>lightrag-data</code>	LightRAG KV store + inputs	~300 MB
<code>lightrag-kg</code>	Knowledge graph files	small
<code>lightrag-cache</code>	LLM response cache	small
<code>lightrag-neo4j-data</code>	Neo4j graph entities + relations	~170 MB

Typical total: **500 MB - 1 GB** compressed.

How it runs (POST-MIGRATION)

Source: Azure VM `vm-alai-lightrag` (20.240.61.67)

- SSH to Azure VM: `ssh -i ~/.ssh/azure_alai alai-admin@20.240.61.67`
- `docker compose stop lightrag neo4j` — graceful shutdown (~30s downtime)
- `docker run alpine tar czf` dumps each volume on VM
- `docker compose start neo4j lightrag` — resume
- `shasum -a 256 *.tar.gz > MANIFEST.sha256` on VM
- Write `README.md` with restore procedure on VM
- SCP from VM to Mac Studio** — download snapshot to `~/system/backups/lightrag/` (safety net)
- Azure offsite upload** — Cool tier blob `plockfrontstaging/lightrag-backup/<TS>/`
- Azure rotation** — keep last 8 snapshots (longer offsite retention)

10. **Local rotation** — keep last 4 snapshots in `~/system/backups/lightrag/` (7-day safety, then deletable)

Downtime: ~60-90s every Sunday 04:00 (cloud LightRAG unavailable during backup).

Key change: Local Docker volumes are NO LONGER the source of truth. Azure VM volumes are primary. Local backups are now safety net only.

Why NOT `docker compose pause`

`pause` freezes LightRAG's async event loop. On unpause, uvicorn stays "running" but HTTP handler doesn't service new requests (container reports `unhealthy`). Requires full container restart to recover. The backup on 2026-04-18 hit this — backup itself was fine (volumes at rest during pause), but container needed `restart` afterwards. Switched to `stop/start` for future runs.

Azure storage details

- Account: `plockfrontstaging` (swedencentral, Hot storage account)
- Container: `lightrag-backup`
- Resource group: `plock-staging-rg`
- Tier per blob: **Cool** (cheaper — ~\$0.01/GB/month for archived reads)
- Retention: last 8 snapshots (~8 weeks)
- Estimated cost: ~**\$0.05-0.10/month** for ~4 GB retained

Restore procedure

Restore to Azure VM (primary, production)

```
# On Mac Studio: pick snapshot
SNAPSHOT=~/system/backups/lightrag/20260418-085317
cd "$SNAPSHOT"

shasum -a 256 -c MANIFEST.sha256 || { echo "checksum mismatch, abort"; exit 1; }

# SCP to Azure VM
scp -i ~/.ssh/azure_alai -r "$SNAPSHOT" alai-admin@20.240.61.67:/tmp/restore/

# SSH to Azure VM
ssh -i ~/.ssh/azure_alai alai-admin@20.240.61.67
```

```
# On Azure VM:
cd /tmp/restore/${basename "$SNAPSHOT"}
shasum -a 256 -c MANIFEST.sha256 || { echo "checksum mismatch, abort"; exit 1; }

cd ~/lightrag
docker compose down

for vol in lightrag-data lightrag-kg lightrag-cache lightrag-neo4j-data; do
    docker volume rm $vol || true
    docker volume create $vol
    docker run --rm -v $vol:/dst -v /tmp/restore/${basename "$SNAPSHOT"}:/src alpine tar xzf
/src/${vol}.tar.gz -C /dst
done

docker compose up -d

# Verify
curl http://localhost:9621/health
# From Mac Studio:
curl https://lightrag.basicconsulting.no/health
```

Restore to Mac Studio (rollback/emergency only)

Use case: Azure VM failure, need to restore local LightRAG as emergency fallback.

```
cd ~/system/docker/lightrag
docker compose down

# Pick a snapshot (local or download from Azure first)
SNAPSHOT=~/.system/backups/lightrag/20260418-085317
cd "$SNAPSHOT"
shasum -a 256 -c MANIFEST.sha256 || { echo "checksum mismatch, abort"; exit 1; }

for vol in lightrag-data lightrag-kg lightrag-cache lightrag-neo4j-data; do
    docker volume rm $vol || true
    docker volume create $vol
```

```
docker run --rm -v $vol:/dst -v "$SNAPSHOT":/src alpine tar xzf /src/${vol}.tar.gz -C /dst
done

cd ~/system/docker/lightrag
docker compose up -d

# Verify
curl http://localhost:9621/health

# IMPORTANT: Update consumer files to use localhost:9621 instead of cloud endpoint
# (see azure-lightrag-migration.md rollback procedure)
```

Azure Blob restore (download offsite backup)

Use case: Local backups lost, need to restore from Azure Blob offsite storage.

```
source ~/system/config/azure-lightrag-backup.env
TS=20260418-085317
RESTORE_DIR=~/system/backups/lightrag/azure-restore-$TS
mkdir -p "$RESTORE_DIR"
az storage blob download-batch \
  --account-name $AZURE_STORAGE_ACCOUNT \
  --account-key "$AZURE_STORAGE_KEY" \
  --source $AZURE_STORAGE_CONTAINER \
  --destination "$RESTORE_DIR" \
  --pattern "$TS/*"

# Verify checksums
cd "$RESTORE_DIR/$TS"
shasum -a 256 -c MANIFEST.sha256

# Then follow "Restore to Azure VM" or "Restore to Mac Studio" procedure above
```

Monitoring

- **Log:** `~/system/logs/lightrag-backup.log` (on Mac Studio, backup orchestrator)
- **Latest snapshot size (local):** `du -sh ~/system/backups/lightrag/`
- **Latest snapshot size (Azure VM):** `ssh -i ~/.ssh/azure_alai alai-admin@20.240.61.67 'du -sh ~/lightrag-backups/'`
- **Azure blob list:**

```
source ~/system/config/azure-lightrag-backup.env
az storage blob list \
  --account-name $AZURE_STORAGE_ACCOUNT \
  --account-key "$AZURE_STORAGE_KEY" \
  --container-name $AZURE_STORAGE_CONTAINER \
  --prefix lightrag-backup/ \
  -o table
```

- **Post-run LightRAG health:** logged as last line of each run (should show `{"status": "healthy"}` from `https://lightrag.basicconsulting.no/health`)

Manual run

```
bash ~/system/tools/lightrag-backup.sh
```

Same 60–90s downtime applies. Log goes to same file.

Note: Post-migration (2026-04-18), script must be updated to SSH to Azure VM instead of using local Docker. See script comments for SSH-based backup procedure.

Related Runbooks

- **Azure LightRAG Migration:** [azure-lightrag-migration.md](#) — full migration context, rollback procedure
- **Ollama Cloudflare Tunnel:** [ollama-cloudflare-tunnel.md](#) — tunnel that LightRAG uses for inference

Document Owner: Skillforge

Last Updated: 2026-08-19 (MC #107355 local recovery and backup restart hardening)

Validated By: Kelsey Hightower (FlowForge), Martin Kleppmann (CodeCraft — data consistency)

Revision #7

Created 2026-04-18 13:44:24 UTC by John

Updated 2026-08-19 21:47:34 UTC by John