

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

# STATUS

site/packages/zapier-decisions/STATUS.md

## Current status — 2026-09-12

Verae Time × Zapier is a **working lab** on one Proxmox host (**NS1.GEORGELAMBERT.ORG**, **70.88.205.138**). Public HTTPS doors, catalog, keep/fleet, and a **test 3-node NATS cluster** on separate LXC guests are up. The test environment **points at that cluster**. Zapier Platform **push** and live **api.veraetime.net** are still waiting on operator login / bind credentials.

**Progress repo:** this file in **zapier-decisions**.

**Catalog:** <https://zapier.georgelambert.org/packages/zapier-decisions/STATUS.pdf>

**Monorepo:** **master-zapier-plan-draft** (**main** / **master**).

---

## Live right now

| Layer                                | State                                                                                                                                                                               |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Catalog                              | <a href="https://zapier.georgelambert.org/">https://zapier.georgelambert.org/</a> (PDF links path-relative; module/model books work)                                                |
| Public doors                         | portal, api, zap, iam, mw, fleet, git — last off-box GET 8/8 OK                                                                                                                     |
| Keep + fleet                         | Running. systemd + linger <b>enabled</b> for next reboot (current processes were not double-started as systemd)                                                                     |
| Test NATS                            | <b>3-node JetStream on vmbr1</b> : LXC 511/512/513<br>nats-a/b/c at 10.10.10.21–23 . Cluster name verae , 2 routes each.                                                            |
| Product streams                      | <a href="#">ZAPIER_JOBS</a> , <a href="#">ZAPIER_EVENTS</a> , <a href="#">ZAPIER_WEBHOOKS</a> ,<br><a href="#">ZAPIER_USAGE</a> , <a href="#">VERAE_ARCHIVE</a> — <b>replicas=3</b> |
| Modules on that cluster              | Fleet, keep, middleware ( <a href="#">natsConnected: true</a> ),<br>billing, access planes. <a href="#">NATS_URL</a> lists all three URLs.                                          |
| Host loopback :4222                  | Still listening; <b>clients no longer use it</b>                                                                                                                                    |
| Worker CT <b>510</b> verae-px-worker | 10.10.10.20 on vmbr1 . Keep worm+tree <b>200</b> on the cluster                                                                                                                     |
| Backup                               | Timer <a href="#">verae-backup.timer</a> enabled; snapshots under <a href="#">/SSD2/backups/verae/</a>                                                                              |
| Uptime                               | Loop on px-worker ( <a href="#">:3870/status</a> ), last tick failed: 0                                                                                                             |

Client URL (private [vmbr1](#) only):

```
nats://10.10.10.21:4222,nats://10.10.10.22:4222,nats://10.10.10.23:4222
```

Repo: <https://git.georgelambert.org/marchon/verae-nats-cluster>

## Cluster message speed ( 20260912T045131Z )

Measured from LXC **510** (not a nats-\* server) against **10.10.10.21-23**.

**Full report (charts + explanations):** [HTML](#) · [PDF](#) · [markdown](#)

**Second study (all code on NS1.GEORGELAMBERT.ORG / 70.88.205.138, run 20260912T051237Z):** [HTML](#) · [PDF](#) · [markdown](#). Orchestrator, nats bench, charts, HTML, and PDF ran on 138; traffic was LXC 510 → 511–513 on **vmb1**. Includes full methodology and fine-tuning notes. Host load ~8.5/40 during the run; JetStream 1p 128 B was **7.4k** durable pubs/s (vs ~16k on the quieter first ladder); ping p99 **1.38 ms**.

**Maximized study ( 20260912T053120Z ):** 8 cores + 16 GiB on 510–513, **max\_mem: 8G**, JetStream **tmpfs** for the ladder (ZFS restored after). veth/10G unchanged. [HTML](#) · [PDF](#).

| Metric                 | Baseline 051237Z | Maximized       | Ratio                     |
|------------------------|------------------|-----------------|---------------------------|
| JS file r=3 1p 128 B   | 7,393 pubs/s     | <b>17,388</b>   | <b>2.35×</b>              |
| JS memory r=3 4p 128 B | —                | <b>36,355</b>   | ~4.9× vs baseline file 1p |
| Core 1p 128 B pub      | 502,502          | 599,004         | 1.19×                     |
| Ping p99               | 1.377 ms         | <b>0.684 ms</b> | <b>2.01× faster</b>       |

Live cluster **keeps 8 cores / 16 GiB / max\_mem 8G**. Product streams are back on ZFS (tmpfs was study-only).

**Optimal-config study ( 20260912T055851Z ):** same 8c/16G cluster on **ZFS**, plus r=1 vs r=3, memory vs file, reconnect tax, UDP echo, MQTT QoS0. Combined report with DL360 Gen10 projection: [HTML](#) · [PDF](#).

| Finding                       | Number                                             |
|-------------------------------|----------------------------------------------------|
| JS file r=3 1p 128 B (8c ZFS) | 14,330 pubs/s                                      |
| JS file r=1 1p                | 18,888 (r=3 costs ~24% on same host)               |
| JS memory r=1 4p              | <b>64,923</b> (lab JS ceiling)                     |
| MQTT QoS0                     | 44,862 pubs/s (not durable; ~15× slower than core) |
| UDP p99 510→511               | 0.363 ms vs NATS ping p99 1.14 ms                  |
| Reconnect-per-msg p99         | 1.75 ms vs persistent ping 1.14 ms                 |

**Do not switch the fabric to MQTT or UDP.** Optimal lab: file r=3 for jobs/archive, memory r=3 optional for events, persistent NATS connections. Metal: 3× DL360 Gen10, local NVMe, 10GbE — projected JS file r=3 ~**40–80k** 128 B pubs/s (not measured).

Short tables: [BENCH.md](#). Re-run: `bash scripts/bench.sh` on NS1.

| Kind                   | Load            | Result                     |
|------------------------|-----------------|----------------------------|
| Core NATS              | 1p1s 50k×128 B  | ~1.20M msgs/s aggregate    |
| Core NATS              | 8p8s 200k×128 B | ~2.01M msgs/s aggregate    |
| Core NATS              | 4p4s 50k×1 KiB  | ~630k msgs/s (~616 MB/s)   |
| JetStream file r=3     | 1–4 pubs, 128 B | ~16k durable pubs/s        |
| JetStream file r=3     | 4 pubs, 1 KiB   | ~13.5k durable pubs/s      |
| Ping (sequential RTT)  | 1k×128 B        | avg 0.307 ms, p99 0.734 ms |
| Flood (burst queueing) | 8p 20k×128 B    | avg 454 ms                 |

Product job/event/archive traffic is the JetStream row. Ping is one-message delay; flood is the mailbox catching up after a burst.

## Built and checked in (Forgejo, each its own repo)

IAM, portal on access-web, keep/watch/guard, fleet floors, host-deps, catalog host-move docs, plus:

| Repo               | Job                                                    |
|--------------------|--------------------------------------------------------|
| verae-uptime       | Off-box HTTPS GET of public doors                      |
| verae-backup       | SQLite / keep / worm-tree tar; restore dry-run default |
| verae-deploy       | Tagged checkout + npm ci on the target OS              |
| verae-proxmox      | LXC 510 worker provision                               |
| verae-nats-cluster | LXC 511–513 JetStream cluster                          |

## Still open

| Item                              | Notes                                                                                                                                                                                                                                 |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Zapier Platform <code>push</code> | Needs Zapier login (see below)                                                                                                                                                                                                        |
| <code>MOCK_VERAЕ=false</code>     | Live <code>api.veraetime.net</code> + admin bind credentials (see below)                                                                                                                                                              |
| NATS nkeys/mTLS                   | Cluster is up <b>without</b> accounts so existing <code>NATS_URL</code> s work. <code>veraе-nats-accounts</code> remains the sketch. Flip only after every client has credentials. Isolation today is <code>vmbr1</code> , not nkeys. |
| lan-134                           | Still disabled (SSH not working)                                                                                                                                                                                                      |
| Reboot test                       | systemd should own keep/fleet after reboot; do not also <code>nohup guard.sh</code>                                                                                                                                                   |
| Hardware NATS                     | Same three configs, three boxes, private NIC — change IPs in <code>cluster.env</code>                                                                                                                                                 |

## Once we have a Zapier login

Zapier cloud only talks **HTTPS** to our Zap plane ( `https://zap.zapier.georgelambert.org` ). No NATS, no `api.veraetime.net`.

1. `zapier login` (or `zapier-platform login`).
2. In `packages/veraе-zapier` (and `veraе-activate` if the tiny app is still wanted):
  - `zapier-platform register` (or attach an existing app id).
  - `MIDDLEWARE_BASE_URL=https://zap.zapier.georgelambert.org` (or `https://mw.zapier.georgelambert.org` if skipping the Zap plane).
  - `zapier-platform build` and `validate`.
3. `zapier-platform push` as a **private** app.
4. Invite the workspace; create a Zap:
  - Auth = portal **API key** ( `zmw_...` ), not a Verae JWT.
  - Test **Create Timestamp and Wait, Create Timestamp** (async `jobId`), **Find Job Status, Timestamp Completed** REST Hook.
5. Fix anything `validate` / `push` flags (auth labels, samples, 402/403 mapping).
6. When the contract is stable, bump the platform version and invite a real user.

Do **not** put NATS URLs or Verae passwords into Zapier.

## Once `MOCK_VERAE=false` and live `api.veraetime.net`

Middleware already has the live HTTP client. Still needed:

1. `VERAE_API_BASE_URL=https://api.veraetime.net` and working **login** (username/password → JWT). JWT stays in middleware ( `tokenRef` ); Zapier never sees it.
2. Confirm OpenAPI vs our client:
  - `POST /api/timestamp` → **202** { `jobId` }
  - `GET /api/status/{jobId}` until `completed` / `failed`
  - `POST /api/verify`
  - Batch create/status if used
3. **Hash lookup is not on the live OpenAPI** — `veraClient.lookupHash` returns **501** when not mocking. Central-chain “find by SHA-256” needs a Verae API gap fill or our own store.
4. Map live status JSON into the receipt shape the Zap app expects ( `certificate` , `timestamp` , `blockIndex` , `sha256` ).
5. Token refresh, 401 retry, and 402/403 mapping against real errors.
6. One live seal in the lab, then REST Hook delivery to `hooks.zapier.com`.

## Submit → `jobId` → wait → receipt

**Yes — that is the designed path**, and it already runs in **mock**.

1. Zap (or curl) `POST /zapier/v1/timestamp` or `/timestamp/wait`.
2. Middleware `createTimestamp` calls Verae `POST /api/timestamp` (or the mock) and gets `jobId`.
3. It publishes `vera.zapier.jobs.watch` on NATS (the 3-node cluster).
4. **Wait:** subscribe to `vera.zapier.jobs.events` for that `jobId`, or poll `GET /api/status/{jobId}`.
5. Job-poller hits Verae status until `completed`.
6. Event `timestamp.completed` carries status; webhook-deliver POSTs the Zapier REST Hook.
7. Response includes **certificate** / **timestamp** / **blockIndex** (mock uses `mock-cert-{jobId}`, `blockIndex: 42`). Async create returns { `jobId`, `sha256`, `existing` } immediately; wait returns the finished status. Same SHA-256 is idempotent ( `existing: true` ).

If wait times out, middleware returns `pending` + `jobId` so the hook can finish the Zap. Live Verae must match that HTTP contract; we have **not** proven a real chain seal yet ( `MOCK_VERAE` is still true).