

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

01 nats routes

site/packages/dev-overview/01-nats-routes.md

Designed vs running NATS routes

Designed = named in docs or in code.

Running = a process on NS1 is subscribed or publishing it **today**, and (for durable work) a JetStream stream exists.

Cluster name `verae`. Internal clients use the three LXC on `vmbr1`. Public edge is a **leaf**, not a second cluster.

Legend: **JS** = JetStream stream; **RR** = core request-reply; **pub** = core or JS fire-and-forget; **Q** = queue group.

Transport (TCP / leaf / public)

These are NATS **server** routes, not `verae.*` subjects.

Endpoint	Designed	Running	Who
10.10.10.21-23:4222 client	yes	yes	middleware, access-*, billing, worm/tree, identity, CS/sales/acct
10.10.10.21-23:6222 cluster routes	yes	yes (private vubr1 only)	nats-a/b/c to each other
10.10.10.21-23:7422 leaf hub	yes (public-edge)	yes	public edge remotes
127.0.0.1:8222 monitor on each LXC	yes	yes	operators on the guest
70.88.205.138:4222 public client	yes (token)	yes	leaf edge; not Zapier
127.0.0.1:14422 WS on edge	yes	yes	Caddy → wss://nats.zapier.georgelambert.org
127.0.0.1:8229 edge monitor	yes	yes	host loopback
:6222 / :7422 / :8222 on vubr0	no	must stay closed	—
Host leftover 127.0.0.1:4222 (docker-proxy) + 10.10.10.1:4222 socat	lab leftover	present	do not point product NATS_URL here
MQTT :1883	study only	removed	—

Internal keep/fleet/middleware URL:

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

Public (token in /etc/nats/public-edge.token on NS1): nats://<token>@70.88.205.138:4222 . Isolation on the hub is still \$G + private LAN. nkeys/mTLS sketched in verae-nats-accounts , **not** flipped.

JetStream streams (running)

Created by `verae-nats-cluster/scripts/ensure-streams.sh` (middleware `ensureStreams` also creates the first three).

Stream	Subjects	Replicas	Storage	Who consumes
ZAPIER_JOBS	<code>verae.zapier.jobs.watch</code>	3	file	job-poller (in middleware process), Q <code>job-poller</code>
ZAPIER_EVENTS	<code>verae.zapier.jobs.events</code>	3	file	middleware wait + webhook router (<code>event-webhook-router</code>); optional exclusive <code>jobs-events</code>
ZAPIER_WEBHOOKS	<code>verae.zapier.webhooks.deliver</code>	3	file	webhook-deliver (in middleware), Q <code>webhook-deliver</code>
ZAPIER_USAGE	<code>verae.zapier.usage</code>	3	file	no durable consumer (<code>USAGE_WRITER</code> named only)
VERAE_ARCHIVE	<code>verae.archive.></code>	3	file	worm + tree on LXC 510 (keep); aggregator not fleet-enabled

Billing is **core request-reply / pub**, not a JetStream stream.

Address inventory

Address	Kind	Designed	Running	Notes
<code>verae.zapier.jobs.watch</code>	JS work Q <code>job-poller</code>	yes	yes	middleware publishes; in-pro consumes
<code>verae.zapier.jobs.events</code>	JS events	yes	yes	poller publishes; wait() subscri worker routes
<code>verae.zapier.webhooks.deliver</code>	JS work Q <code>webhook-deliver</code>	yes	yes	same Node process as HTTP
<code>verae.zapier.usage</code>	JS optional	yes	stream yes , consumer no	middleware entitlement is lo
<code>verae.internal.jobs.events</code>	core republish	yes (<code>jobs-events</code> exclusive)	partial	only if <code>JOBS_EVENTS_EXCLUS</code>
<code>verae.billing.statement.get</code>	RR Q <code>account-balance</code>	yes	yes	portal, CS, sales, accounting,
<code>verae.billing.balance.get</code>	RR	yes	yes	account-balance
<code>verae.billing.balance.adjust</code>	RR	yes	yes	CS credits, portal reload
<code>verae.billing.usage.recorded</code>	pub	yes	yes if <code>edge NATS_URL</code> set	<code>zapper/src/app.ts</code> onUsa <code>natsPublish</code> ; account-balan
<code>verae.billing.payment.recorded</code>	pub	yes	yes if <code>NATS_URL</code>	portal reload (<code>portal.ts</code>)
<code>verae.billing.credit.applied</code>	pub	yes	yes if <code>NATS_URL</code>	admin/CS (<code>admin.ts</code>)
<code>verae.billing.customer.put</code>	pub	yes	yes if <code>NATS_URL</code>	<code>ledger.ts</code> ; account-balance
<code>verae.access.authz.check</code>	RR Q <code>access-authz</code>	yes	yes	fleet keepFloor
<code>verae.access.authz.deny</code>	pub audit	yes		no SIEM sink

Address	Kind	Designed	Running	Notes
			yes (on deny)	
verae.access.web.in / verae.access.web.billing.*	ingress	yes	yes	mapped after authz, e.g. <code>verae.access.web.bill</code>
verae.access.api.*	ingress	yes	yes	
verae.access.zapier.*	ingress	yes	yes	HTTPS plane; Zapier never s
verae.access.leaf.in / verae.access.leaf.*	ingress	yes	yes	no billing
verae.access.staff.*	ingress	yes	yes	CS/sales/accounting
verae.identity.bind	RR Q identity	yes	yes	fleet
verae.identity.lookup	RR Q identity	yes	yes	
verae.archive.put	JS (VERAE_ARCHIVE)	yes	partial	splitter in-process in middl on 510 consume
verae.archive.query	broadcast (no Q)	yes	partial	live aggregator hook TBD in (simulator has it)
verae.archive.reply.<id>	replies on bloom hit	yes	partial	worm/tree on 510 can reply; enabled
verae.splitter.in	future	yes (docs)	no	splitter stays a library
verae.example.process.in / .out / template	template	yes	no	<code>verae-nats-process</code> is a m
bench.*	lab only	n/a	ephemeral	speed studies; not product

Plane mapping (`verae-access-authz`): internal `verae.billing.statement.get` + plane `web` → ingress `verae.access.web.billing.statement.get`. Planes never publish internal subjects until `authz.check` returns allow.

Durable consumers (named in middleware)

Name	Stream / subject	Running
job-poller	ZAPIER_JOBS / jobs.watch	yes (in-process)
event-webhook-router	ZAPIER_EVENTS / jobs.events	yes (in-process)
webhook-deliver	ZAPIER_WEBHOOKS / webhooks.deliver	yes (in-process)
usage-writer	ZAPIER_USAGE / verae.zapier.usage	no
jobs-events-exclusive	ZAPIER_EVENTS	no unless JOBS_EVENTS_EXCLUSIVE=1

Fleet overlay vs in-process

packages/verae-fleet/overlays/ns1/fleet.json (2026-09-14):

Service	fleet <code>enabled</code>	Actual lab
middleware-http	true	running ; starts job-poller + webhook workers in-process when <code>NATS_ENABLED</code>
job-poller	false	covered by middleware; do not spawn a second copy
webhook-deliver	false	same
archive-aggregator	false	not running as a fleet worker
archive-worm / tree-node	false on host fleet	running on LXC 510 via keep <code>units.px-worker.json</code> (1 worm + 1 tree)
account-balance, access-*, identity, jobs-events, CS/sales/acct, staff-iam/session, zappier-edge	true keepFloor	keepFloor
zapier-simulator	true, keepFloor false	optional
nats	managed false	3-node cluster + leftover host loopback