

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

NATS.IO 3-server cluster

site/packages/overview/83-nats-cluster.md

3. Central Verae NATS.IO 3-server cluster

All durable messaging for this product is **NATS JetStream**, not Zapier queues and not a public TCP API.

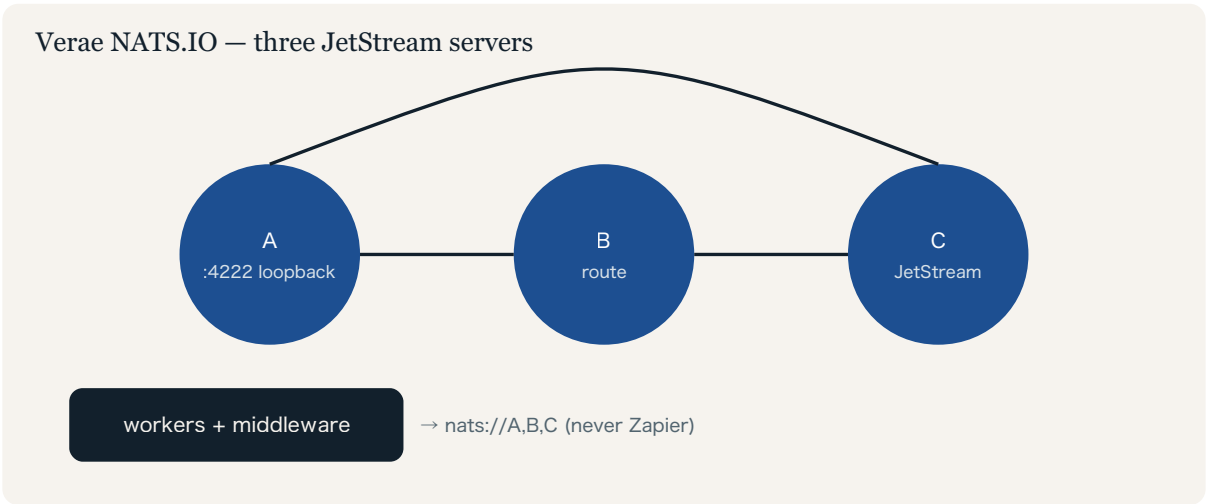
Cluster

Three `nats-server -js` nodes form the Verae cluster (routes between them, JetStream replication). Clients (middleware, poller, aggregator, WORM, tree nodes, account-balance, zappier-edge billing, CS/sales/accounting, and any new process cloned from `verae-nats-process`) connect with a **cluster URL list**, for example:

```
nats://127.0.0.1:4222,nats://127.0.0.1:4223,nats://127.0.0.1:4224
```

On a given machine the listener stays on **loopback** (or a private interface). Operators reach it with `scripts/nats-tunnel.sh` / `ssh -L 14222:127.0.0.1:4222`. **Do not bind 4222 on 0.0.0.0 without auth.**

Today's NS1 box (`NS1.GEORGELAMBERT.ORG`, `70.88.205.138`) already runs JetStream on `127.0.0.1:4222`. The **target** is three clustered nodes so losing one server does not lose the stream. Fleet SSH hosts (`ns1`, later `lan-134`) run **workers**, not extra public NATS listeners.



NATS cluster

Who may connect

Allowed	Forbidden
verae-middleware, fleet workers, WORM, tree nodes, account-balance, zappier-edge billing, CS/sales/accounting, <code>verae-nats-process</code> clones	Zapier cloud, customer browsers (they use HTTPS portal/CS/sales UIs)

Address families already in use

Address	Kind	Notes
<code>verae.zapier.jobs.watch</code>	work queue	poller
<code>verae.zapier.jobs.events</code>	events	wait + webhooks
<code>verae.zapier.webhooks.deliver</code>	work queue	HTTPS to Zapier REST Hook
<code>verae.zapier.usage</code>	optional	metering
<code>verae.billing.statement.get</code>	request-reply	portal, CS, sales, admin
<code>verae.billing.balance.adjust</code>	request-reply	CS credits, portal reload
<code>verae.billing.usage.recorded</code>	pub	zappier-edge meter
<code>verae.archive.put</code>	JetStream	splitter / merkle builder
<code>verae.archive.query</code>	broadcast (no queue group)	every WORM and tree node
<code>verae.archive.reply.<correlationId></code>	replies	only on bloom hit

New functions get new `verae.<area>.<resource>.<action>` addresses — see [o6-address-routing.md](#).