Ops Journal — issues, root causes, and fixes

Ops Journal — issues, root causes, and fixes

A running log of things that broke, *why* they broke, how they were fixed, and what to do so they never happen again. Newest entry first. Each entry follows the same shape so it stays useful as a reference, not just a diary.

> Maintained by Claude across sessions. When a real issue is found and fixed, add an entry here — symptom, root cause, fix, and the prevention that makes it non-recurring. Keep it concrete: exact files, commands, and IPs.

Template:

```
## YYYY-MM-DD — <short title>
**Symptom / how it surfaced:**
**Root cause:**
**Fix (what was actually done):**
**Prevention (so it never happens again):**
**Lessons / notes:**
```

2026-11-05 — vps → vps2 migration (headscale, marketdata,

strata; mail staged)

Symptom / how it surfaced: Not an incident: `vps` (`203.57.114.42` / `100.64.0.2`) is being deleted, so everything it did had to move to `vps2` (`187.52.126.150` / `100.64.0.16`) without breaking sites or mail.

Root cause: The old box was doing three unrelated jobs — public TLS edge, mail ingress (HAProxy), and outbound SMTP relay — plus running headscale.

Fix (what was actually done):

- **headscale + headplane**: `docker compose stop` on the vps, re-synced `headscale/lib` (db.sqlite + `noise_private.key`) and `headplane/lib` while the container was stopped (a hot copy leaves a 2.2 MB WAL behind), started them on vps2. Nodes kept their keys and 100.64.0.x addresses — no re-registration. Both bind `127.0.0.1`; Caddy terminates TLS.
- **marketdata**: rsync of the web tree, `mysqldump`/reload of the 3.2 GB DB (42 tables), php8.4-fpm + MariaDB with a 1 GB buffer pool, and the Python pipeline (uv, playwright chromium 1161, HotCopper CDP profile, cron `0 5 * * 2-6`).
- **strata**: rsync of `/home/conor/Documents/strata-project` (excluding `.git`, `node_modules`, `data/backups`, then GNAF separately), npm install, PM2 with both apps (`strata` 3009, `strata-demo` 3011) bound to `127.0.0.1`.
- **Ansible**: vps2 Caddyfile is now the whole edge; new `roles/postfix_relay`; vps2 added to the `edge` group; mail ports opened in `host_vars/vps2.yml`. Commit `c1edc4f`.

Prevention (so it never happens again):

- `ufw.service` can read `inactive (dead)` while the ruleset is loaded and enforced. Check `sudo ufw status verbose`, never `systemctl is-active`, before concluding a port is open or that a provider is filtering.
- Hostinger does **not** block port 25. The "filtered" result on mail ports was ufw's default deny.
- After any `rsync -a` into `/var/www`, re-run `chown -R conor:caddy <site>`. rsync overwrote the provisioner's ownership and PHP could not read the tree.
- Postfix bound to a Tailscale address must wait for it at boot, or it dies with `fatal: parameter inet_interfaces: no local interface found` and stays dead. `roles/postfix_relay` ships the wait helper + `Restart=on-failure`.

Lessons / notes:

- The `mailcow` role had been silently doing nothing: `mailcow_dir` pointed at the deleted lowercase `/home/conor/docker/mailcow`. Its `relayhost` also lived only in the live `extra.cf`, so the first real run would have removed the outbound relay. Both fixed in the same commit.
- rsync is the right tool for a one-shot move; Syncthing (bidirectional daemon) is not, because it propagates deletions.
- Direct cbfarm → vps2 transfer does not work from a background shell on cbcore (`/home/conor/.ssh/server: No such file or directory`, no agent context). The two-hop cbfarm →

cbcore → vps2 path ran at 1.5–4 MB/s and is simpler.

2026-08-16 — mailcow open relay: ~94k spam messages sent from our IP

Symptom / how it surfaced: Abusix network-abuse report, forwarded by BinaryLane, naming `203.57.114.42` (the vps, `mail.seabee.me`) as a spam source. Investigation found the server had been an **open relay** from 2026-08-05 15:52 UTC to 2026-08-13 18:06 UTC: 94,223 messages accepted, 93,602 delivered. Spoofed senders `test@seabee.me` (82k), `info@seabee.me` (11k), and MetaMask phishing. **Not a compromise** — no breached accounts, no rogue SASL logins; the spam never authenticated.

Root cause: Three things lined up:

1. mailcow trusts its own Docker subnet — `mynetworks` included `172.29.1.0/24`.
2. Port 25 was published on the Tailscale IP and handled by Docker's **userland proxy** (`docker-proxy`), which reopens each connection *from the bridge gateway* `172.29.1.1`. So every external sender appeared to come from `172.29.1.1`.
3. `172.29.1.1` is inside the trusted `/24` → `permit_mynetworks` → relay granted to anyone. HAProxy passthrough was NOT the bug; the bug was source-IP masking turning an untrusted sender into a trusted one.

It stopped by accident on 2026-08-13 when a container restart recreated the DNAT rules (so source IPs were briefly preserved again). The hole was still live — verified a forged connection from `172.29.1.1` got `250 Ok` for an external recipient until fixed.

Fix (what was actually done):

- `data/conf/postfix/extra.cf`: rewrote `mynetworks` to cover only real container addresses (`172.29.1.2`–`.255`) and **exclude the gateway** `172.29.1.1`: `127.0.0.0/8 [::1]/128 172.29.1.2/31 172.29.1.4/30 172.29.1.8/29 172.29.1.16/28 172.29.1.32/27 172.29.1.64/26 172.29.1.128/25`. mailcow re-injects this into the regenerated `main.cf`. Verified: relay from `172.29.1.1` now `454 denied`; legit inbound + internal relay still work.
- **Rotated the DKIM key** for `seabee.me` (selector `dkim`) — the spam went out DKIM-signed as us (`sign_local=true`, `use_domain=envelope`), so domain reputation was hit, not just the IP. New key published + cryptographically verified against mailcow's private key.
- **Purged 35,567** stuck spam messages from the vps Postfix queue (`postsuper -d` on senders `test@`/`info@`/`noauth@seabee.me`), stopping the retry storm against GMX/web.de

that was deepening the IP throttle.
- Fixed vps **system-mail routing** (was `myorigin=/etc/mailname=mail.seabee.me`, which is mailcow's name; set `myhostname/mailname=vps.seabee.me`, `mydestination` local, `root: conor` alias) so cron/root mail delivers.
- **Restored spam filtering:** enabled PROXY protocol end-to-end. HAProxy on the vps now uses `send-proxy-v2` to mailcow's haproxy-enabled ports (`10025/10465/10587/10993/14190`); mailcow again sees the **real client IP** (verified: DNSBL/`zen.spamhaus.org` and rDNS now evaluate the true sender). HAProxy is now Ansible-managed (`roles/haproxy` in machine-config).

Prevention (so it never happens again):

- **Never trust the Docker bridge gateway (`172.29.1.1`) in `mynetworks`.** The new list fails safe: if source IPs are ever masked again, mail is *rejected*, not relayed.
- **PROXY-protocol ports let the connecting host assert any client IP** — so they are firewalled to the vps only via `~/mailcow-proxy-guard.sh` (a `DOCKER-USER` rule that self-heals on reboot; `selftest` proves a non-vps tailnet source is dropped). Never publish those ports without that guard.
- **Set `{"userland-proxy": false}`** in `/etc/docker/daemon.json` on cbcore so a forwarded connection's source IP can never again be rewritten to an internal address. (Recommended; confirm it's applied.)
- **DMARC is still `p=none`** — that let the spoofing pass unchallenged. Move to `p=quarantine` now DKIM is confirmed working.
- **No outbound rate limit** existed — nothing throttled the 94k. Add a per-hour cap as a damage limiter so any future compromise/relay is bounded to hundreds, not tens of thousands.

Lessons / notes:

- Running any MTA behind a proxy/tunnel makes **source-IP masking** a first-class risk: it can silently convert `permit_mynetworks` into an open relay. Audit what address the MTA actually *sees* for external connections, not what you assume.
- **I lost the Aug 5–13 postfix logs by recreating the container before exporting them.** Next time: export/copy container logs *before* any recreate. mailcow's redis log buffer only holds ~18h.
- The **vps had no persistent mail logging** (`rsyslog` inactive, Postfix not logging to journald) — which is why message *bodies* were never recoverable. Fix vps logging before the next incident.
- Chosen architecture (settled 2026-08-16): mailcow **at home** (data + TLS key never leave cbcore), vps as a **blind TLS-passthrough pipe** + outbound relay. This is the privacy-optimal design given CGNAT. The WireGuard+relay migration was considered and rejected (relay would hold plaintext on the vps).

Revision #7
Created 6 November 2026 12:14:29 by bot
Updated 6 November 2026 13:15:46 by bot