

Post-Migration Checklist: Legacy → Docker

Post-Migration Checklist: Legacy → Docker

Run these steps **after** `scripts/migrate_legacy_to_docker.sh` completes. The migration restores the databases and portable data (encryption key, TLS certs, DKIM signing keys), migrates admin users to LDAP, and — at the end of its run — invokes `system_rehost.sh` to rewire the host identity and re-render the Docker-managed service configs. This checklist covers what remains.

“ **Scope:** validated for migrating from legacy build **240815** (relay topology) to the current Docker release. The migration script rejects any other source build.

“

⚠ Outbound email is PAUSED until you release it

The migration deliberately brings the gateway up with **all outbound delivery held**. A freshly migrated box has the restored quarantine and its notification schedule live — left alone, it will blast **stale quarantine notifications to every real recipient** in the restored data (this is not hypothetical; it will happen). So nothing leaves the box: outgoing mail parks in the queue.

Work through this checklist, then [release outbound as the very last step](#) — after you've reviewed the queue and deleted anything you don't want sent. Inbound mail and filtering run normally the whole time; only egress is held.

What the migration already did for you

- Restored `hermes` (schema-forwarded to the current baseline), `djigzo`, `opendmarc`, `Syslog`; created `authelia` and `nextcloud`.
- Copied portable data: `hermes.key` (DB-secret encryption key), Let's Encrypt certs, DKIM signing keys.
- Migrated Authelia users into LDAP as **one-factor** admins.
- Ran `system_rehost.sh`, which rewired host identity (`server_ip`, `console.host`, Postfix `myhostname/myorigin`, `nginx server_name`, Authelia cookie domain) and **re-rendered postfix, amavis, authelia, nginx, and mailname** from the restored database, then restarted those containers.

Settings Postfix reads through **MySQL maps — Domains, Relay Recipients, Virtual Recipients**, sender/recipient rules — are live directly from the restored database and need no action.

“ If you skipped host rewiring during the migration, run it now: `sudo scripts/system_rehost.sh --force` (auto-detects this host's IP/hostname; override with `--to-hostname=` / `--to-ip=` if needed).

1. Log in and sanity-check

1. Log into the admin console (see step 4 if all admins show as one-factor).
2. Sidebar → **Domains** and **Relay Recipients** — confirm the migrated data.
3. Sidebar → **System Status** — confirm all containers are healthy.

2. Regenerate mail authentication (DKIM, SPF, DMARC)

`system_rehost.sh` does **not** cover these. Each regenerates its own service config from the database; their `TrustedHosts` lists must carry the Docker subnet. The DKIM signing keys are already migrated (`opt/hermes/dkim/keys`).

All three live under the **Content Checks** section of the sidebar.

1. **Content Checks → DKIM Settings → Save & Apply Settings** — regenerates OpenDKIM KeyTable / SigningTable / TrustedHosts referencing the migrated keys. Confirm each domain's published DNS TXT record still matches.
2. **Content Checks → SPF Settings → Save & Apply Settings** — regenerates policyd-spf.
3. **Content Checks → DMARC Settings → Save & Apply Settings** — regenerates OpenDMARC.

3. Verify Relay Networks (mynetworks)

1. Sidebar → **Relay Networks** — confirm the customer's allowed sending networks carried over **and** the Docker subnet is present; **Save & Apply** if you change anything.
2. Sidebar → **Relay Host** — verify the upstream smarthost and credentials; re-enter the relay password if the field is blank.

4. Reconfigure admin two-factor authentication

Every migrated admin is in the **one-factor** group; legacy TOTP / WebAuthn devices were not carried over (Authelia moved SQLite → MySQL).

1. Verify each admin can log in with their **existing (legacy) password**.
2. Sidebar → **System Users** — move each admin who should use 2FA to the **two_factor** group; have them re-enroll an authenticator on next login.

Important: review every admin account — all are password-only until you do this.

5. Reactivate the Pro license (Pro editions)

The Pro license is bound to the server's hardware UUID, which changed on the new Docker host, so it shows **invalid / pending**.

1. Email support with the license **serial number** to request reactivation.
2. Support deactivates the old activation so it can re-bind to the new host.
3. Reactivate, then log out and back in to re-validate.

Important: until reactivated, the system runs as Community Edition.

6. Verify TLS certificates

The Let's Encrypt certificates were copied to the Docker cert path.

1. Sidebar → **System Certificates** — confirm the console and mail certificates are present and selected (not the `bootstrap_hermes.pem` placeholder).
2. Confirm HTTPS (console) and SMTP STARTTLS work.

7. Email archive (deferred)

Message History **metadata** is restored, but message **bodies** live in the separate archive backup (`hermes-archive-<build>-...tar.gz`). Restore it when convenient — it needs no gateway downtime.

- Sidebar → **Content Checks** → **Message History** — entries appear; bodies are viewable only after the archive is restored.

8. Final end-to-end test

1. Send a test message **inbound** and **outbound**.
2. Verify: DKIM = pass, spam scoring, and TLS on both legs.



Outbound is still **paused** at this point, so an outbound test message **parks in the queue** rather than delivering. That's expected — you confirm actual outbound delivery in step 9 after releasing.

9. Release outbound delivery

This is the last step. Do it only after everything above checks out.

While migrating, outbound has been held so the box could not mail real recipients (see the notice at the top). Now release it — deliberately, after reviewing what's queued.

1. Sidebar → **Mail Queue**. At the top, the **Outbound Delivery** card shows **PAUSED**.
2. **Review the queued mail**. Expect a large batch of quarantine-notification messages (`from postmaster@...`) generated from the restored quarantine. **Select and delete** those — they are stale and should not go out. Keep any genuine mail you want delivered.
3. Click **Resume Outbound Delivery**. This clears the hold and flushes the queue; the **Outbound Delivery** card flips to **ACTIVE**.
4. Confirm the outbound test from step 8 now delivers, and watch the mail log to be sure only the mail you kept goes out.

“ The **Outbound Delivery** control (Mail Queue page) is a permanent pause/resume switch — useful for maintenance windows too. Pausing holds all outgoing mail in the queue without affecting inbound or filtering.

Revision #8

Created 2026-07-20 13:31:48 UTC by Dino Edwards

Updated 2026-08-22 11:48:22 UTC by Dino Edwards