Get Started (Docker)

Get Started (Docker)

This page is the **minimum config needed to get mail flowing** on a fresh Hermes SEG Docker install. The install script (`scripts/install_hermes_docker.sh`) does most of the heavy lifting; this page covers the handful of admin-UI steps that still need a human.

Skip these and Postfix will silently bounce or reject mail. The admin dashboard also surfaces two universal nudges (placeholder hostname, self-signed cert) until those are addressed (see Dashboard nudges at the bottom).

Which steps apply to you?

Hermes supports three deployment topologies. **Step 1 (System Identity) and the Optional/DNS sections apply to everyone**, and **Step 2 (Console FQDN and a real certificate) is required for anything that uses Nextcloud**. The middle of this guide then splits into a **Relay** path and a **Mail server** path. Follow only the one(s) for your topology:

Topology	What it is	Follow
Relay-only	Hermes filters mail and forwards it to a downstream mail server (MX)	Step 1 → **Relay configuration**
Mail-server-only	Hermes hosts the mailboxes itself (Dovecot + webmail)	Step 1 → **Step 2** → **Mail server configuration**
Hybrid	Both: some domains relay out, others have local mailboxes	Step 1 → **Step 2** → **Relay configuration** → **Mail server configuration**

> **Legacy reference**: this page replaces the pre-Docker 16-step page. The Docker install script absorbs ~6 of those steps, so the list below is shorter.

What the install script already did

You don't need to redo any of this; `install_hermes_docker.sh` handled it during the install run:

Component	Result
Containers	All Hermes containers running (`docker compose ps`)
Bootstrap admin	LDAP user in `cn=admins` + `cn=one_factor`, password in `INSTALL_SUMMARY`
TLS	Self-signed bootstrap cert in System Certificates, bound to Console / SMTP / Webmail roles
Databases	MariaDB schemas (hermes, djigzo, opendmarc, syslog, authelia, nextcloud) created + seeded
Console settings	`parameters2.console.host` set to the **host IP**, not your FQDN. See the note below
Postfix identity	`myhostname` / `myorigin` set from the install-time mail-hostname prompt
Authelia	LDAP backend wired up; 2FA enrollment available on first login
Mail filtering	Amavis, SpamAssassin and ClamAV running. Network checks and Bayes still need setup. See Antispam Settings

> **Why the console address is an IP.** At install time there is usually no DNS record yet for the FQDN you intend to use, so the installer deliberately points the console, Nginx, Authelia and Nextcloud at the host IP. That way you can log in immediately. Once DNS resolves, save **System → Console Settings** with your FQDN and Hermes re-renders the whole web stack. Do this before handing the console to anyone else.

So **after the install you can log in, but mail won't actually flow** until you complete the steps below.

Step 1: System Identity (all topologies)

Page: System → Server Setup

The install script sets `myhostname` from what you typed at the FQDN prompt, but you should double-check it matches your DNS A / MX records. Also set:

- **Postmaster address**: where bounce messages and admin notifications go
- **Admin email**: where alerts (license, system events) get delivered
- **Time zone**: affects log timestamps and report scheduling

> **Dashboard nudge**: an orange callout `Placeholder hostname` fires (any topology) if `myhostname` still equals the seed default `hermes.domain.tld` or `console.host` equals `smtp.domain.tld`. Both should never appear on a Docker install (the install script overrides them), but if they do, this is the page to fix.

Step 2: Console FQDN and a real certificate

Required for mail server and hybrid. Optional for relay-only.

Pages: System → Console Settings, then System → System Certificates

The installer leaves you with a console reachable at the host IP, secured by a self-signed bootstrap certificate whose common name is `localhost`. That is deliberate, and it is enough to log into the admin console and enough for mail to flow.

It is not enough for Nextcloud. Until you finish both parts of this step, `https://<console-host>/nc/` login fails for every user, including the administrator. Webmail, the Nextcloud Mail client, and file access are all unreachable until it is done.

Why it fails

Nextcloud signs users in through OIDC. That is not a browser-only flow: the Nextcloud container makes its own server-side HTTPS call back to the console address to reach the identity provider. On a fresh install that call is refused, because the bootstrap certificate is self-signed and its common name is `localhost`, which matches neither the host IP nor the FQDN you intend to use.

Importing the bootstrap certificate into Nextcloud's trust store **does not fix this**. Trust and name are two separate checks, and the name still does not match. The only fix is a certificate issued for the console's real name.

Part 1: point the console at an FQDN

Create a DNS A record for the name you want (for example `mail.example.tld`) pointing at this host, and wait for it to resolve. Then open **System → Console Settings**, enter that FQDN, and save. Hermes re-renders Nginx, Authelia and Nextcloud, and restarts the web stack.

The save is gated on DNS: if the name does not resolve to this host, Hermes keeps the current address rather than locking you out.

Part 2: install a real certificate for that FQDN

Go to **System → System Certificates** and issue or import a certificate covering the FQDN you just set. The three paths are described under Real TLS Certificate below.

The certificate must cover the **console address specifically**. A certificate valid for your mail domain but not for the console name leaves Nextcloud login broken in exactly the same way.

Confirm it worked

Open `https://<your-fqdn>/nc/` and log in as any mailbox user. A successful sign-in means the OIDC round trip completed, which is the thing that was failing.

If you see "Could not reach the OpenID Connect provider", one of the two parts is incomplete: either the console is still on an IP, or the certificate does not cover the name in use.

Part 3: bind the certificate to SMTP as well

This is a separate setting and it is the one people miss. Setting the Console Certificate does **not** configure Postfix. They are two independent bindings:

Role	Page	Serves
Console Certificate	System → Console Settings	nginx on 443: `/admin`, `/users`, `/nc`
SMTP TLS certificate	**System → SMTP TLS Settings**	Postfix on 587 and 465

Go to **System → SMTP TLS Settings**, select the same certificate you bound to the console, and save.

Skip this and the console looks perfect in a browser while **every mail client gets a certificate error**, because Postfix is still serving the install-time bootstrap certificate with common name `localhost`. Thunderbird, Outlook and phones either refuse the connection or train the user to click through a warning every time.

It has to be the certificate covering the **console host**, because that is the name clients are told to use: `autoconfig.cfm` reads `console.host` and hands it out as both the IMAP and the SMTP server, and the SRV records on the Mailbox Domains page point at the same host.

Confirm it took:

```
docker exec hermes_postfix_dkim postconf -n | grep smtpd_tls_cert_file
```

It should name your real certificate. If it still says `bootstrap_hermes.pem`, the setting was not saved. A browser check cannot tell you this, because the browser never connects to Postfix.

> **Relay-only deployments** do not use Nextcloud, so the Nextcloud half of this step is not blocking. The certificate still matters if your users connect any mail client. Do it before handing the console to anyone else either way, so administrators are not training themselves to click through certificate warnings.

Relay configuration

*For **relay-only** and **hybrid** deployments. If Hermes hosts your mailboxes and never forwards to a downstream MX, skip this whole section and follow **Mail server configuration** below.*

A. Relay Domains

Page: Email Relay → Domains

Add **at least one domain** so Hermes knows what mail to accept on the SMTP port. Without this, every inbound message gets rejected with `Relay access denied`.

For each domain you'll choose:

Field	What it controls
Domain	The recipient domain (e.g. `example.com`)

Field	What it controls
Recipient delivery mode	Where validated mail goes next: **relay** forwards to a downstream MX (the usual relay-topology choice)
Destination address / port	The downstream MX host + port that accepts the forwarded mail
Policy	Encryption policy applied to outbound mail for this domain (Pro only)

B. Relay Networks

Page: Email Relay → Relay Networks

Add the IP addresses or CIDR blocks of any **upstream MTA** (your customer's mail server, an application server that sends notification mail, etc.) that should be allowed to relay outbound mail through Hermes.

By default Hermes only trusts `127.0.0.1` and the Docker bridge subnet (`172.16.32.0/24`). Anything else needs to be added here.

C. Relay Recipients

Page: Email Relay → Relay Recipients

Add the individual recipients (or wildcards) that Hermes should accept mail for. Validated mail is then forwarded to the destination set on the domain row in step A. Without at least one recipient, mail for the domain is rejected as unknown.

Mail server configuration

*For **mail-server-only** and **hybrid** deployments. If Hermes only relays to a downstream MX and hosts no mailboxes, skip this whole section.*

A. Mailbox Domains

Page: Email Server → Domains

Add **at least one mailbox domain**, the domain Hermes will host mailboxes for. This is a *different* page from Email Relay → Domains: it provisions the local-delivery side (Dovecot,

autoconfig/autodiscover, webmail), not relay forwarding.

When you add a mailbox domain Hermes sets up the per-domain mail-client autoconfiguration. For TLS, mailbox domains need a certificate that also covers `autoconfig.<domain>` and `autodiscover.<domain>`. See **Real TLS Certificate** below.

B. Mailboxes

Page: Email Server → Mailboxes

Create the individual mailboxes under your mailbox domain(s). Each mailbox row creates an LDAP user, a Dovecot maildir, and (optionally) a Nextcloud account for webmail/file access. Users log in to webmail via Authelia SSO at `https://<console-host>/nc/`.

Optional but recommended

Relay Host (Outbound Smarthost) *(relay / hybrid)*

Page: Email Relay → Relay Host

If outbound mail should route through an upstream provider (Gmail, Microsoft 365, SendGrid, etc.) instead of being sent directly to recipient MXes, configure the smarthost here. Authentication credentials are encrypted at rest using the Hermes install's key material.

Pro License Activation *(all topologies)*

Page: System → Server Setup → License section

Enter your serial number to unlock Pro features (organizational signatures, encrypted mail, ARC sealing, Link Guard, etc.). Validation hits `validate.hermesseg.io` over HTTPS; the result is cached locally so Pro stays available during brief network outages.

Real TLS Certificate *(all topologies)*

Page: System → System Certificates

Replace the bootstrap self-signed certificate with a real one before going live. On mail server and hybrid installs this is **not optional**: Nextcloud login cannot work until the console has an FQDN and a certificate issued for it. See Step 2. Three paths:

Path	Tier	Workflow
Request ACME	Both tiers	Click → enter domain → Let's Encrypt issues automatically, auto-renews
Import Certificate	Both tiers	Paste cert + key + chain from any CA you already have
Generate CSR	Both tiers	Generate signing request → submit to CA → import the result via the **Import Certificate** path

For mailbox-hosting domains, see the in-app "Choosing the Right Certificate Type" panel on the System Certificates page; mailbox certs need SAN coverage for `autoconfig.<domain>` and `autodiscover.<domain>`.

> **Dashboard nudge**: blue informational callout `Self-signed cert` fires when the only row in `system_certificates` is the install-generated bootstrap (no real cert has been imported yet). Mail flows on the bootstrap certificate and clients get a TLS warning, but on mail server and hybrid installs the consequence is larger than a warning: **Nextcloud login does not work at all** until a real certificate covering the console FQDN is in place. See Step 2.

DKIM Signing *(all topologies)*

Page: Content Checks → DKIM Settings

A fresh install has no DKIM keys, and that is expected. Keys are per-domain, so they cannot exist before your domains do. Until you generate one, `KeyTable`, `SigningTable` and the `dkim_sign` table are all empty and `/opt/hermes/dkim/keys` holds nothing. OpenDKIM is running correctly the whole time; it simply has nothing to sign with.

The order matters:

1. **Add the domain first**: Email Relay → Domains, or Email Server → Domains
2. **Generate the key**: Content Checks → DKIM Settings, select the domain, choose a selector and key size
3. **Publish the TXT record** the page gives you at your DNS provider
4. **Verify**: the page re-checks the published record and reports a match

Repeat per domain. Outbound mail for a domain is not signed until step 3 has propagated.

DNS for Mail Flow *(all topologies)*

Beyond the gateway itself, DNS is what makes mail actually arrive. The install script does **not** touch DNS; you do this at your registrar. "Your domains" below means relay domains, mailbox domains, or both, whichever you configured above.

Record	Where it points	Why
`MX` for each domain	The Hermes mail hostname (e.g. `mail.example.com`)	Inbound mail routing
`A` for the mail hostname	Hermes public IP	Resolves the MX target
Reverse DNS (PTR) for the IP	The mail hostname	Outbound deliverability; most receivers reject mismatched PTR
`SPF` for each sending domain	Includes Hermes IP	Authenticates outbound; reduces spam-folder rate
`DKIM` selector → public key	Generated under Content Checks → DKIM Settings	Cryptographic signing of outbound
`DMARC` policy	TXT at `_dmarc.example.com`	Defines what receivers do with SPF/DKIM failures

Review the Admin Account Email *(all topologies)*

Page: System → System Users (edit the admin user), or your **My Profile** link (top of the sidebar)

The install created the admin account with a generated email of the form `<admin-username>@<your-mail-domain>` (e.g. `apologise4567@example.com`). That address is where Hermes sends **admin notifications and password-reset mail**, so unless it maps to a real, monitored mailbox, change it to one that does.

Antispam Settings (Pyzor / Razor / Bayes) *(all topologies)*

Page: Content Checks → **Antispam Settings** (both the on/off switches and the one-time actions)

Spam filtering works out of the box from SpamAssassin's rule set, RBLs, ClamAV and the malware feeds. Three optional components need a deliberate decision from you.

Collaborative network checks (Pyzor, Razor, DCC)

These check a message's fingerprint against a shared network of reporters. They are effective, and they **transmit a digest of every message you scan to a third party**. That is a decision for you, not for us.

- **From v260807 they ship disabled.** Enable them under **Antispam Settings** if you want them.
- **On installs predating v260807** they were enabled but never registered, so they contributed nothing. Enabling them properly is the same procedure.

Once enabled:

Check	What it needs
Pyzor	Nothing. It works as soon as it's enabled (outbound internet required).
Razor	A one-time registration via **Content Checks → Antispam Settings → Initialize Razor**. Until this runs, Razor returns no result.
DCC	Not included in the published image for licensing reasons. See Antispam Settings if you want to add it.

Bayes

The Bayesian classifier learns what *your* mail looks like. It ships **empty** and contributes **no score at all** until it has learned roughly 200 spam and 200 ham messages. That is SpamAssassin's own safety threshold, not a Hermes limitation.

- Train it with **Train as Spam** / **Train as Ham** in Quarantine and Message History. Users can do this from their own portal too.
- **Automatic learning is off by default and we recommend leaving it off.** Autolearn trains on whatever the rule set already decided, so it reinforces the rules' mistakes as readily as their successes, and on a gateway that sees mostly spam it skews badly.
- **Installs predating v260807** carry a Bayes corpus that shipped with Hermes, trained on unrelated mail. The v260807 upgrade clears it once so your gateway learns from your own traffic.

Barracuda Central Registration *(all topologies)*

Hermes' Postfix `postscreen` DNSBL list includes `b.barracudacentral.org`, and Barracuda Central only answers queries from **registered** IPs. Register your gateway's sending IP (free) at the Barracuda

Reputation Block List site so those lookups return results instead of being silently ignored.

CipherMail Console Admin Password *(all topologies (encryption))*

Page: the CipherMail console at `/ciphermail` (behind Authelia SSO)

The CipherMail encryption console has its **own** administrator account, separate from the Hermes/Authelia admin login. It ships with CipherMail's stock default credentials:

Username	Password
`admin`	`admin`

Sign in with those and change the password immediately. The account is not managed by Hermes, so nothing else will prompt you and no dashboard nudge fires for it. Authelia SSO gates the `/ciphermail` path, which means the default is not reachable from the internet, but it remains a stock credential on an admin interface and should not survive your first login.

Things that look broken but aren't

A fresh install leaves several things deliberately empty. If you go looking under the hood before configuring anything, these are the ones that reliably cause alarm.

What you see	Why	What to do
No DKIM keys. `KeyTable`, `SigningTable` and `dkim_sign` are empty; `/opt/hermes/dkim/keys` holds nothing	Keys are per-domain and can't precede your domains	Generate them after adding a domain
MariaDB root has no password. `docker exec hermes_db_server mariadb -u root` connects with no credentials	`root@localhost` uses the **`unix_socket`** plugin, so it only trusts a process already running as root inside the container. The password in `INSTALL_SUMMARY.txt` belongs to `root@'%'`, the remote entry, and port 3306 is not published to the host	Nothing. This is the intended design
The `migrations` table is empty	It records one-time upgrade migrations. A fresh install has never upgraded, so there is nothing to record	Nothing

What you see	Why	What to do
Bayes reports no data / no effect on scores	It ships empty and stays inert until ~200 spam + 200 ham are learned	Train it
`ecprivkey.pem` **and** `ecpubkey.pem` **are empty files**	Mailbox encryption is off by default. The placeholder files exist only so Docker doesn't create directories in their place	Enable it under Email Server → Settings, which generates the real keypair. **Back the keys up**: losing them makes encrypted mail permanently unreadable

Nextcloud login needs the console FQDN and a real certificate first

Before anything below matters, the console must be on an FQDN with a certificate issued for it. On a fresh install neither is true, and `/nc/` login fails for everyone with "Could not reach the OpenID Connect provider". This is the single most common reason webmail appears broken on a new deployment. See Step 2.

Trusting the bootstrap certificate inside the Nextcloud container does not help, because its common name is `localhost` and will never match the console address.

Nextcloud must be enabled when you create the mailbox

Enable Nextcloud on the Add Mailbox form, not afterwards. The setting controls three things at once: `/nc` access, the Nextcloud account itself, and the Nextcloud Mail profile that authenticates to Dovecot on the user's behalf.

The default comes from the domain's own Nextcloud setting, which is off unless you turned it on when you added the domain, so it is easy to leave off without noticing.

If Nextcloud Mail shows no account, or reports an authentication failure, that is the cause. **Do not try to add the account by hand with the user's login password**. It will never work, because the login password is not a mail credential. Hermes authenticates IMAP and SMTP only against app passwords, and Nextcloud Mail uses an automatically managed one the user never sees. Thunderbird and phones use app passwords the user generates in the portal under **My App Passwords**.

> Installs predating v260807: enabling Nextcloud *after* a mailbox was created did not provision the Mail profile. The v260807 upgrade repairs affected mailboxes automatically.

Dashboard nudges

The admin dashboard surfaces two universal callout banners under the navbar. These apply regardless of topology:

Color	Priority	Trigger
Orange `Placeholder hostname`	2	`myhostname` or `console.host` still at the seed placeholder (`hermes.domain.tld` / `smtp.domain.tld`)
Blue `Self-signed cert`	3	Only the bootstrap cert exists in System Certificates; no real cert imported yet

Each banner links directly to the page where you'd fix the underlying condition and disappears automatically as soon as the check passes. The topology-specific steps (relay domains, networks, recipients, mailbox domains, mailboxes) intentionally don't have dashboard nudges; they depend on which topology you're using and the guidance above covers them.

After you finish these steps

1. **Inbound test**: send a message from an external account to a recipient on one of your domains. Check Reports → Mail Log to confirm it reached Hermes and was handed off (relay) or delivered to the mailbox (mail server).
2. **Outbound test** *(relay / hybrid)*: send a message from your customer MTA (the one whose IP you added to Relay Networks) to an external recipient. Confirm DKIM/SPF pass on the receiver side.
3. **Webmail test** *(mail server / hybrid)*: log in to `https://<console-host>/nc/` as one of your new mailbox users (Authelia SSO) and confirm send/receive. If this fails with "Could not reach the OpenID Connect provider", Step 2 is incomplete: the console is still on an IP, or the certificate does not cover the name in use.
4. Open the **Admin Console** home page and confirm both setup nudges are gone (placeholder hostname + self-signed cert).
5. If you set up Pro features, verify `session.edition` reads "Pro" in the top-right corner of any admin page.

You're done. Welcome to Hermes SEG.

Revision #38
Created 2026-05-31 13:01:43 UTC by Dino Edwards
Updated 2026-08-22 11:48:52 UTC by Dino Edwards