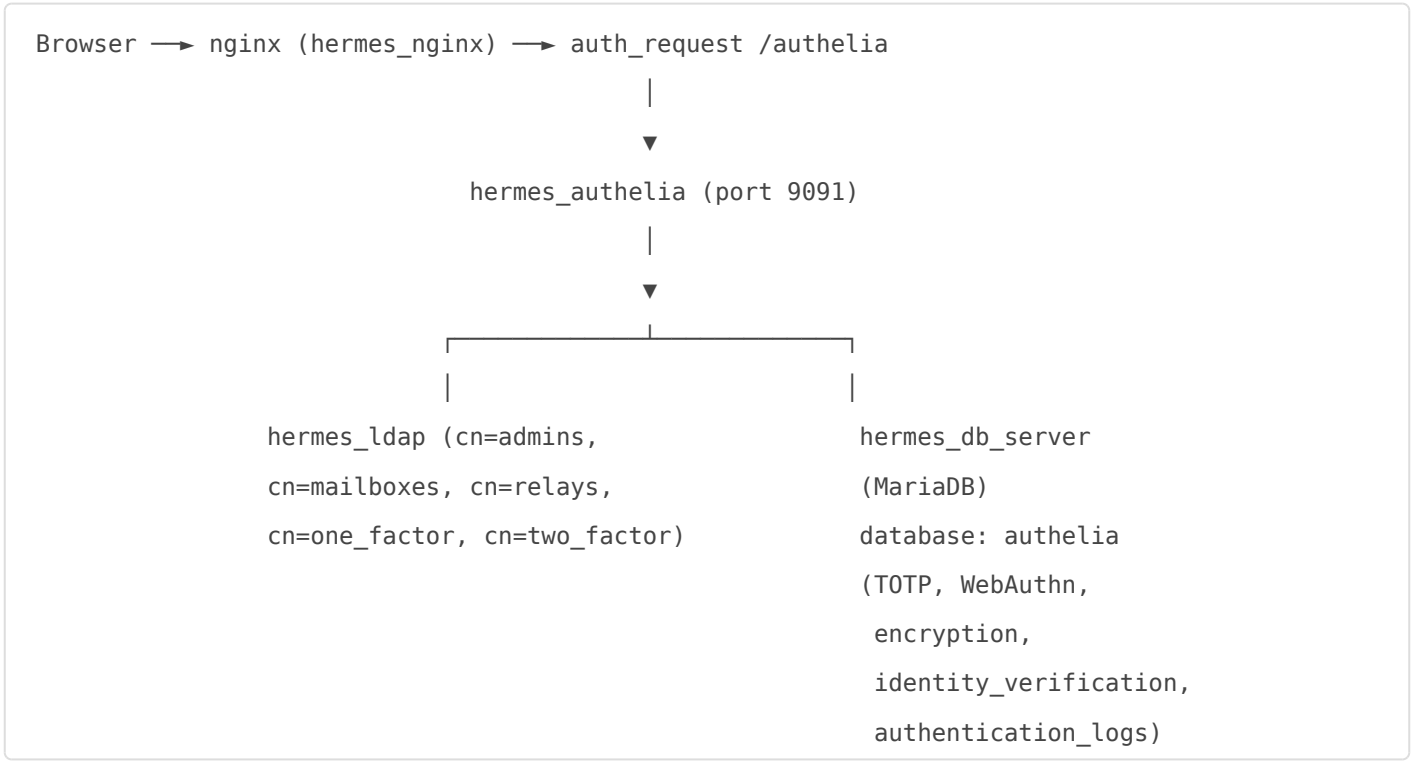
Authentication Settings

Authentication Settings

Admin path: **System > Authentication Settings** (`view_authentication_settings.cfm`, `inc/get_authelia_settings.cfm`, `inc/edit_authelia_settings.cfm`, `inc/auth_generate_secret.cfm`, `inc/generate_authelia_configuration.cfm`, `inc/restart_authelia.cfm`).

This page configures **Authelia** — the identity-aware proxy that gates every Hermes web surface (`/admin`, `/users`, `/nc`). It is global gateway plumbing: secrets, session timing, login-failure regulation, SMTP notifier credentials, Duo Push integration, and the OIDC client that Nextcloud uses for SSO. Per-user MFA enforcement, app passwords, and the local-vs-remote credential model are documented in the Credential Model chapter and on the LDAP RemoteAuth page; this page is strictly the gateway configuration.

Where Authelia sits



Every protected request triggers nginx's `auth_request /authelia` which proxies to `hermes_authelia:9091/api/verify`. Authelia checks its session cookie (stored in Redis via `hermes_authelia_redis`), and if needed redirects the user through a one-factor (password) or two-factor (password + MFA) login flow against the LDAP directory. The nginx snippets that wire this up are `config/nginx/etc/nginx/snippets/auth.conf` and `snippets/authelia.conf`.

The Authelia container reads its config from `/config/configuration.yml` inside the container (host path `config/authelia/configuration.yml`). The config file is **regenerated from a template** every time this page is saved — the template at `/opt/hermes/templates/configuration.yml` (host path `config/authelia/configuration.HERMES`) is read, the `hermes_*` placeholders are substituted with values from `parameters2` where `module = 'authelia'`, the result is written, and the container is restarted. Direct edits to `configuration.yml` are overwritten on the next save.

Configuration storage and persistence

Setting class	Lives in	Read by
Toggles, durations, hostnames, log level	`parameters2` table, `module = 'authelia'`	Form load via `get_authelia_settings.cfm`; template substitution at regen
High-entropy secrets	Files under `/opt/hermes/keys/` (Docker secret mounts)	Authelia reads via `{{ secret "..." }}` directives in `configuration.yml`
Sessions (cookie state)	Redis (`hermes_authelia_redis`)	Authelia at runtime
MFA registrations	MariaDB `authelia` database	Authelia at runtime; encrypted at rest with the Storage Encryption Key
Identity verification tokens	MariaDB `authelia` database	Reset-password and add-device flows

Secrets are never round-tripped through the form. Read-only fields on the page show a masked tail (last 4 chars) of the file contents so the admin can verify the secret exists and roughly recognise it. The regenerate button next to each field writes a fresh random value directly to disk (`auth_generate_secret.cfm`), regenerates `configuration.yml`, and restarts Authelia.

Storage backend — MySQL, not SQLite

Authelia stores MFA registrations, identity-verification tokens, and audit logs in the `authelia` MariaDB database on `hermes_db_server`. This is intentionally different from Authelia's upstream SQLite default:

- **Survives container recreation.** Docker `down`/`up` cycles wipe named volumes if the operator hasn't bind-mounted SQLite's storage path. MariaDB lives on the Data tier and is backed up by the standard system backup.
- **Tolerates concurrent reads.** SQLite serialises writes; with hundreds of mailboxes hitting `/users` and `/nc` simultaneously this becomes a contention point.
- **Single backup surface.** The Hermes system backup already includes all MariaDB databases. The Authelia DB is included automatically; no separate path to remember.

The credential to the `authelia` database is stored as a Docker secret file at `/opt/hermes/keys/authelia_db_password` and referenced from the Authelia config via `{{ secret "/keys/authelia_db_password" | msquote }}`.

Cards on the page

General Settings

Field	What it controls	Stored as
Password Reset JWT Secret	Signs the time-limited token in password-reset email links. Rotating invalidates every outstanding reset link.	File `/opt/hermes/keys/authelia_identity_validation_reset_password_jwt_secret_file`
Reset Password Function	Enable/disable the "Forgot password?" link on the login page. Disable when password is owned by remote AD/LDAP.	`parameters2.authentication_backend.disable_reset_password`
Storage Encryption Key	AES key Authelia uses to encrypt TOTP secrets and WebAuthn credentials at rest inside the `authelia` database. **Rotating this key invalidates every TOTP and WebAuthn registration** — every MFA-enrolled user must re-enrol on next login.	File `/opt/hermes/keys/authelia_storage_encryption_key_file`

> **Do not rotate the Storage Encryption Key casually.** The red callout on the page exists for a reason. Rotation is correct after a confirmed compromise; in every other case it locks every MFA user out of their tokens. Duo Push survives

> because Duo enrollment lives in Duo's cloud, not the Authelia DB — see the Duo section below.

Session Settings

Field	Default	Notes
Session Name	`hermes_session`	Cookie name. Changing forces every active session to log in again.
Session Secret	random	Signs the session cookie. Rotating invalidates all sessions immediately.
Session Provider Password (Redis)	random	Auth between Authelia and `hermes_authelia_redis`. Rotating requires the Redis container to pick up the new secret on Authelia restart.
Session Expiration	`43200` (12h)	Absolute lifetime from login. NIST SP 800-63B AAL2 ceiling.
Session Inactivity	`3600` (1h)	Idle timeout. NIST 800-63B recommends 1800s (30 min) for AAL2 / 900s (15 min) for AAL3.
Remember Me Duration	`43200` (12h)	When ticked at login, replaces Session Expiration **and bypasses Session Inactivity entirely**. Set to `-1` to remove the checkbox from the login form.

The "Remember Me" interaction is the gotcha. Authelia 4.39 source (`internal/handlers/handler_authz_authn.go`) confirms that a remembered session is exempt from inactivity checks — its lifetime is the Remember Me Duration, full stop. If your compliance posture requires inactivity enforcement on **every** session, set Remember Me Duration to `-1`; otherwise users who tick the box are governed only by the absolute ceiling.

Saving this card also pushes matching values into Nextcloud via `occ config:system:set` (`session_lifetime`, `session_keepalive`, `remember_login_cookie_lifetime`). NC sessions are kept in lockstep with Authelia to prevent stale NC sessions from triggering the OIDC auto-redirect URL mangle.

SMTP Notification Settings

The address and subject Authelia uses when sending password-reset, identity-verification, and new-device-registration emails. Hermes points Authelia at its own internal Postfix re-injection port (`hermes_postfix_dkim:10026`) so notification mail goes through the gateway's outbound pipeline like

any other Hermes-originated message. SMTP host/port are not exposed on this page — they are hard-coded in the template because there's no real reason to change them on a self-contained install.

Login Regulation

Authelia's built-in brute-force throttle.

Field	Effect
Login Failures Before Ban	Number of consecutive failures from one source before that source is banned (default 5)
Time Between Failed Logins	Sliding window over which the failure count is measured, in seconds (default 120)
Banned Time	How long the ban lasts, in seconds (default 300)

This is the inner brake. The outer brake is the Intrusion Prevention `authelia` jail (`hermes_fail2ban`) which scans `/remotelogs/authelia/authelia.log` and applies host-level iptables bans for longer durations. The two layers are complementary: Authelia regulates per-account in the application; Fail2ban regulates per-source-IP at the firewall.

Logging

Authelia log level (`trace`, `debug`, `info`, `warn`, `error`), format (`json` or `text`), and retention in days. The retention dropdown applies to the **rotated** Authelia log files (`config/authelia/log/authelia.log.*`) — the live file is rotated by the host logrotate config and old rotations are pruned to the retention window. Default 30 days; legal/compliance reviewers may need 90 or 180.

Duo Security

Optional second factor via Duo Push (mobile-app one-tap approval). Disabled by default. When enabled, fields are required:

Field	Source
Duo Hostname	Duo Admin Panel → Applications → Auth API → "API hostname" (`api-XXXXXXXX.duosecurity.com`)
Duo Integration Key	Same panel, "Integration key"
Duo Secret Key	Same panel, "Secret key"
Duo Self Enrollment	If enabled, users who don't yet have a Duo account can self-enrol from the Authelia MFA page

Integration and Secret keys are stored as Docker secret files at `/opt/hermes/keys/authelia_duo_api_integration_key_file` and `authelia_duo_api_secret_key_file`. The form blanks the input on display and only writes when a non-empty value is submitted (the masked tail under the box shows the current value's last 4 chars). This lets the admin save other fields without re-entering Duo credentials every time.

> **Duo survives storage-key rotation and SQLite-to-MySQL migrations.** Duo enrollment lives on Duo's servers, not in Authelia's database; Hermes only stores the API credentials. The TOTP and WebAuthn tables in the `authelia` MariaDB database are wiped when the storage key rotates or the SQLite-to-MySQL migration runs; Duo Push keeps working.

Webmail OIDC (Nextcloud)

Authelia acts as the OpenID Connect provider for Nextcloud's `user_oidc` app — this is what makes "Sign in with Hermes" work on `/nc` and (transparently) auto-login users who already have a valid Authelia session.

Field	Role	Stored as
OIDC HMAC Secret	Signs Authelia-issued OIDC tokens	`/opt/hermes/keys/authelia_identity_providers_oidc_hmac_secret_file`
OIDC Client Secret	Shared secret between Authelia (RP) and Nextcloud (client). Hashed with PBKDF2 inside Authelia.	Plain: `authelia_identity_providers_oidc_clients_client_secret_plain_file`; digest: `authelia_identity_providers_oidc_clients_client_secret_digest_file`
OIDC Key	RSA 2048 private key (JWKS) Authelia uses to sign ID tokens	`/opt/hermes/keys/authelia_identity_providers_oidc_jwks_file` (generated with `openssl genrsa`)

The OIDC client is registered as `Hermes_SEG_Webmail`, redirect URI `https://<console>/nc/apps/oidc_login/oidc`, scopes `openid profile email groups`. The `groups` scope is what gives Nextcloud the LDAP group claims it needs to apply NC's own group ACLs.

Rotating the OIDC Client Secret triggers a follow-up `occ user_oidc:provider Hermes_SEG --clientsecret=...` execution against the `hermes_nextcloud` container so both sides stay in sync. Rotating the HMAC Secret or OIDC Key on Authelia's side will invalidate all in-flight OIDC sessions — users will see a fresh login challenge on next request.

What this page does NOT configure

Setting	Lives on
Console hostname (Authelia `session.cookies[].domain` + `authelia_url`)	Console Settings — regenerating console settings re-templates Authelia and restarts it
LDAP backend address / bind DN / filters	Hard-coded in the template to point at `hermes_ldap`. The Hermes LDAP container's structure is provisioned at install time and not exposed as a runtime knob.
Upstream AD / LDAP authentication for specific mailboxes or relay recipients	LDAP RemoteAuth — Authelia still binds locally; the local LDAP entry has a `seeAlso` overlay pointing at the upstream directory
Per-user MFA enforcement	The admin's mailbox/relay-recipient detail pages — `recipients.enforce_mfa` is a TINYINT(3) admin-policy flag (see Credential Model and #225 below)
Password reset flow UI	Password Resets — the reset page itself, CAPTCHA, rate limiting
System users / admins list	System Users — managing accounts in `cn=admins`, `ou=users`
Fail2ban brute-force protection	Intrusion Prevention — the host-firewall layer in front of Authelia
Nextcloud OIDC auto-redirect toggle	Email Server Settings — moved off this page; controls whether `/nc` silently SSOs the already-authenticated user

MFA enforcement is decoupled from the `cn=two_factor` LDAP group (#225)

This is the single most-often-confused part of Hermes authentication.

The LDAP group `cn=two_factor` is a *capability* marker, not an *enforcement* marker. Membership in `cn=two_factor` tells Authelia "this user has at least one MFA method registered and should be prompted for it." Membership in `cn=one_factor` tells Authelia "password only." A user

moves from `one_factor` to `two_factor` by **enrolling an MFA method themselves** through the user portal's Account Settings page — admins do not force-flip the group.

Admin policy lives in `recipients.enforce_mfa` (and `system_users.enforce_mfa` for system users) — a TINYINT(3) column, not a group. When the admin sets this to 1, the user-portal pages that depend on it consult `config/hermes/var/www/html/users/2/inc/check_enforce_mfa_restriction.cfm` on each request. If the user is in `cn=mailboxes` or `cn=relays`, has `enforce_mfa = 1`, and is **not** yet in `cn=two_factor`, the page renders a restricted-access panel pointing them at Account Settings to enable 2FA. Once they enrol, the group flips and the restriction clears on the next page load.

Why this two-layer model

The chicken-and-egg without it: enrolling TOTP or WebAuthn requires the user to receive an identity-verification email from Authelia. A brand-new mailbox has no working mail client yet — they need to get into the portal first to set up an app password, configure their phone, and read the email. Hard-locking them out of the portal until they enrol means they can never enrol.

The bootstrap surfaces (Account Settings, My App Passwords, Set Up Your Devices, Webmail) remain accessible under the restriction; the rest of the portal does not. Once the user enables 2FA, the restriction lifts automatically.

Operational consequence — log out, don't just refresh

Authelia caches LDAP group membership in the session for the refresh interval (default 5 minutes). When a user enables 2FA, their LDAP group flips to `cn=two_factor` immediately, but Authelia's session still says `cn=one_factor` until the cache expires. Hermes works around this by redirecting through `/logout` after the enable-2FA flow, which forces a fresh Authelia session and picks up the new group membership on the next request. If a user reports "I enabled 2FA but the portal still says I haven't," the answer is always: log out and back in.

Save flow

Save & Apply Settings runs `edit_authelia_settings.cfm`, which:

1. Validates every form field (whitelist regex per field, length minimums for secrets).
2. Updates the matching `parameters2` rows with `applied = '2'`.
3. After all field updates succeed, flips every `module = 'authelia'` row to `applied = '1'`.
4. Calls `generate_nextcloud_configuration.cfm`, pushes session parameters into Nextcloud via `occ config:system:set`, and restarts `hermes_nextcloud`.

5. Calls `generate_authelia_configuration.cfm` which re-templates `configuration.yml` from `/opt/hermes/templates/configuration.yml`.
6. Calls `restart_authelia.cfm` (which uses the canonical preload pattern, not a hard restart, to avoid `ERR_CONNECTION_REFUSED` on the redirect back).
7. Sleeps 10 seconds to let Authelia come back up before the redirect lands.

If validation fails at any step the form short-circuits via `cflocation url="#cgi.http_referer#"` with a `session.m` alert code; no partial state is committed because each cascade step gates on `step = N`.

Failure semantics

What breaks	What happens
Authelia container down	nginx `auth_request` returns 500; every protected page shows "502 Bad Gateway" or similar. Mail flow is unaffected — Postfix, Dovecot, and Amavis don't depend on Authelia.
MariaDB `authelia` database unreachable	Authelia starts but cannot authenticate; same symptom as above.
Redis (`hermes_authelia_redis`) down	Authelia starts but cannot store sessions; users are bounced to the login page on every request.
Storage Encryption Key file missing	Authelia refuses to start. Check `docker logs hermes_authelia` for the missing-secret error.
`configuration.yml` syntax-broken after a bad save	Authelia refuses to start. Restore from the on-disk backup `configuration.BACKUP`, fix in the form, save again.
LDAP container down	Authelia starts but every login attempt fails. Same recovery as MariaDB-down — fix LDAP first, no Authelia restart needed.

The Save & Apply Settings button does not have a pre-save dry-run; if Authelia refuses to start after a save, the previous `configuration.yml` is no longer on disk. The `restart_authelia.cfm` step will surface the container start failure in the admin UI's restart-output area; the admin should not navigate away until the success banner appears.

Related documentation

- Credential Model — the four-credential architecture (web login, app passwords, NC internal password, Hermes System app password) that this page's session settings gate
- LDAP RemoteAuth — when web login is bound against an external AD/LDAP instead of Hermes's internal directory
- Password Resets — the forgot-password page that consumes the JWT Secret on this page

- Console Settings — the console hostname change that triggers an Authelia template re-render
- Intrusion Prevention — the Fail2ban `authelia` jail that protects this surface at the firewall
- System Users — admin accounts that live in `cn=admins`
- Email Server Settings — the Nextcloud OIDC auto-redirect toggle that complements the OIDC client configured here

---

Revision #8
Created 2026-05-31 12:51:55 UTC by Dino Edwards
Updated 2026-05-31 14:01:02 UTC by Dino Edwards