Server Setup

Server Setup

Admin path: **System > Server Setup** (`view_server_setup.cfm`, `inc/save_server_identity.cfm`, `inc/generate_postfix_configuration.cfm`, `inc/generate_nextcloud_configuration.cfm`).

This page configures **how Hermes identifies itself to other mail servers** — the Postfix `myorigin` domain, the `myhostname` FQDN used in SMTP banners and HELO/EHLO greetings, and the host IPv4 address used by Nextcloud's `trusted_domains`. These are foundational, mostly install-time values; changing them in production has visible downstream effects on outbound mail acceptance and on email-client configuration.

Pairs with Console Settings, which configures the web-side identity (Console Address and certificate). The two pages together define every name Hermes presents to the world: the mail side on this page, the web side on Console Settings.

What this page does NOT configure

Concern	Lives on
The hostname/IP that nginx terminates HTTPS on for `/admin`, `/users`, `/nc`	Console Settings — Console Address
The TLS certificate presented to mail clients on `:25`, `:465`, `:587`	SMTP TLS Settings — separate cert binding from the console cert
The TLS certificate presented to the web console	Console Settings — Console Certificate
Per-domain mail routing, accepted-domain lists, relay maps	Email Relay > Domains and Email Server > Domains
The Docker subnet (`IPV4SUBNET` in `.env`)	Currently hardcoded in 15+ config files. See Known limitation below.

Concern	Lives on
Initial install — admin password, LDAP base, secrets generation	`scripts/install_hermes_docker.sh` (see Release engineering and updates)

Configuration storage — the `parameters` / `parameters2` split

This page is one of the cleanest examples of the **dual-role `parameters` table** in Hermes. Two of the three fields live there (under their Postfix directive names), and the third lives in `parameters2`.

`myorigin` and `myhostname` — `parameters` table

In the `parameters` table, the same directive is stored as **two rows**:

Row	Role	Linked by
`child = 2` row	The directive **name** (the Postfix keyword), e.g. `parameter = 'myorigin'`	`parent_name` on the value row points back to this row's `parameter`
`child = 1` row	The directive **value** (the actual domain/hostname), e.g. `parameter = 'example.com'`, `parent_name = 'myorigin'`	—

The page reads from the `child = 1` row (the value) and writes back to the same `child = 1` row when an admin saves. The `child = 2` row's `enabled` flag is set to `1` on every save to guarantee the directive is included when Postfix `main.cf` is regenerated.

```
-- The name row (directive)
parameter = 'myorigin', child = '2', enabled = '1', conf_file = 'main.cf', module = 'postfix'

-- The value row (the actual domain)
parameter = '<your-domain>', parent_name = 'myorigin', child = '1',
    module = 'postfix', conf_file = 'main.cf'
```

The same shape applies to `myhostname`. Seeded defaults are `domain.tld` and `hermes.domain.tld` respectively.

> **Why the split.** The dual-row pattern lets Hermes treat any Postfix directive uniformly: the parent (`child = 2`) carries metadata — display name, help text, default, enable flag — and one or more value rows (`child = 1`) carry the actual configuration. Multi-value directives (`mynetworks`, `smtpd_recipient_restrictions`, etc.) just have more `child = 1` rows under the same `parent_name`. Single-value directives like `myhostname` have exactly one.

Host IP Address — `parameters2` table

Host IP lives in `parameters2` because it is not a Postfix directive — it is a free-floating piece of installation state consumed by Nextcloud's `trusted_domains` config.

```
parameter = 'server_ip', value2 = '<ip>', module = 'network'
```

Read by `generate_nextcloud_configuration.cfm` and substituted into `config.php` as `NEXTCLOUD_TRUSTED_DOMAIN_IP`. The same value is also used by the install script and any other code that needs the operator-confirmed host IP without parsing it out of `ip addr`.

Fields on the page

Mail Server Domain (Postfix `myorigin`)

The origin domain Postfix appends to unqualified sender addresses on outbound mail. If a local process submits a message from `root@localhost`, Postfix rewrites it to `root@<myorigin>` before sending. For internal-only setups this can stay at the install default; for any system that sends external mail, set it to the operator's canonical domain.

Validated by the email-trick: `IsValid("email", "test@<value>")` must return true. Empty input is rejected with `session.m = 2`; invalid format with `session.m = 4`.

Mail Server Hostname (Postfix `myhostname`)

The fully-qualified hostname Hermes announces in its SMTP banner and HELO/EHLO greeting. This is the value other mail servers see when they connect to Hermes (and that Hermes presents when it connects to them). Three downstream consequences:

Consumer	What goes wrong if this doesn't match DNS

Receiving MTAs' reverse-DNS checks (PTR lookup → A lookup → match)	Recipient servers reject outbound mail with `450/550 helo not match` errors
TLS certificate Common Name / SAN match on SMTP	Strict STARTTLS verifiers refuse to deliver to Hermes
Authoritative SPF / DKIM / DMARC alignment for `mailfrom`	Indirect — bounces may align poorly if MAIL FROM uses an unmatched domain

> **Do not change this in production without planning.** The page wraps the field in a red warning callout for a reason. The page warning enumerates the user-visible breakages:
>
> - All external email clients (Thunderbird, Outlook, iOS Mail, etc.) need their IMAP/SMTP server hostname reconfigured
> - CalDAV/CardDAV clients need new server URLs
> - Nextcloud Mail profiles for **remote-auth** mailboxes (auto-discovered via the external FQDN) re-prompt for the user's AD password and auto-update on the next login
> - Nextcloud Mail profiles for **local-auth** users are unaffected — those profiles use internal Docker hostnames (`hermes_postfix_dkim`, `hermes_dovecot`), not the external FQDN
>
> Plan the change for a maintenance window, notify users, and have new client setup instructions ready.

Validation: email-trick again (`IsValid("email", "test@<value>")`). Empty → `session.m = 3`; invalid → `session.m = 5`.

After a successful save, also ensure a matching TLS certificate is bound for SMTP on SMTP TLS Settings. The hostname change does not automatically rebind the cert; both must match for STARTTLS handshakes to verify.

Host IP Address

The operator-confirmed IPv4 address of the Docker host. Used to populate Nextcloud's `trusted_domains` so NC accepts requests routed through the IP literally (some autoconfig and CalDAV/CardDAV clients hit the IP before they have the FQDN).

Validation: `^(\d{1,3}\.){3}\d{1,3}$` — basic IPv4 dotted-quad. Empty is allowed (skips the regen of that field). Invalid → `session.m = 6`.

The Host IP and the Console Address are independent. If the Console Address on Console Settings is set to an **IP** (rather than an FQDN) and the host IP changes, you must update both pages — neither cascades into the other. If Console Address is an FQDN, only this page needs the IP update.

Save flow

Clicking **Save & Apply Settings** posts `action=save_settings`, which runs `save_server_identity.cfm`:

```
1. Validate all three fields (presence + format)
2. UPDATE parameters2.value2 WHERE parameter = 'server_ip'
3. UPDATE parameters.enabled = '1' WHERE parameter IN ('myorigin','myhostname')
   AND child = '2' AND module = 'postfix'         (re-arm both directives)
4. UPDATE parameters.parameter = <domain>
   WHERE parent_name = 'myorigin'  AND child = '1' AND module = 'postfix'
5. UPDATE parameters.parameter = <hostname>
   WHERE parent_name = 'myhostname' AND child = '1' AND module = 'postfix'
6. INCLUDE generate_postfix_configuration.cfm   (rewrites main.cf + reload)
7. INCLUDE generate_nextcloud_configuration.cfm (rewrites NC config.php)
8. cflocation back to view_server_setup.cfm with session.m = 1 (success)
```

There is no nginx restart in this cascade — only **Postfix** and **Nextcloud** are touched. That is deliberate: nothing in the nginx-served path consumes `myorigin`, `myhostname`, or the network `server_ip` (the nginx vhosts use the **Console** Address, configured separately). The save flow is therefore much lighter than Console Settings: typically 5–10 seconds, no overlay spinner, no preload-style restart.

`generate_postfix_configuration.cfm` re-templates `config/postfix-dkim/etc/postfix/main.cf` from the live `parameters` rows (walking every `child = 2` row that has `enabled = 1`, emitting each as `<keyword> = <value>` with values pulled from the matching `parent_name`-linked `child = 1` rows), copies the result into the `hermes_postfix_dkim` container, and runs `postfix reload`. The reload is a SIGHUP — it does **not** drop in-flight SMTP connections; mail flow continuity is preserved across the save.

`generate_nextcloud_configuration.cfm` rewrites the entire `config.php` from its template (`/opt/hermes/templates/config.php`), substituting the host IP into `trusted_domains` along with all the other NC settings the regenerator owns. Existing installation-specific values (`passwordsalt`, `secret`, `instanceid`, `version`) are read back from the live file first and preserved — the regenerator never invents new versions of these or NC would think it needs to re-install.

Failure semantics

What breaks	What happens
Validation fails on any field	`session.m = 2..6`, `cflocation` back to the page, no DB write
`parameters` UPDATE succeeds but `generate_postfix_configuration.cfm` fails to write	DB is ahead of the live config. Next save (or any other Postfix-config save) re-regenerates `main.cf` from the same DB rows and catches up.
`postfix reload` fails inside the container	DB and on-disk config are in sync but the running Postfix is still on the old config. Symptom: outbound mail still uses the old `myhostname`. Recovery: `docker exec hermes_postfix_dkim postfix reload` manually, or re-save.
`generate_nextcloud_configuration.cfm` fails (e.g., NC container down)	Postfix change is committed; NC is stale. Recovery: bring NC up and re-save, or re-run the regen include directly.
Hostname change breaks reverse DNS at the recipient	Hermes accepts the change cleanly; the visible failure is **deferred** — outbound mail starts getting rejected by other MTAs minutes to hours later. Always verify PTR + matching A record **before** changing `myhostname`.

The save flow has no rollback. The previous `main.cf` lives at `config/postfix-dkim/etc/postfix/main.cf.HERMES` (the CFML write-time backup convention) and can be restored manually if a regen produces broken syntax — but the DB has already advanced.

Known limitation — Docker subnet is hardcoded

The Docker subnet that Postfix and Amavis trust (`IPV4SUBNET=172.16.32` in `.env`) is **not** managed on this page. It is currently hardcoded into 15+ config files spanning Postfix (`mynetworks`, `master.cf`), Amavis (`@inet_acl`), Dovecot (`login_trusted_networks`), Ciphermail (`authorizedAddresses`), OpenDKIM/OpenDMARC (`TrustedHosts`), and several CFML queries.

If you need to change the subnet for IP-conflict reasons, **all 15+ files must be updated coherently** or mail flow will break in subtle ways (Amavis rejecting messages from Hermes itself, OpenDKIM not signing outbound, etc.). This is a tracked tech-debt item — when templating is added, the subnet will move into `system_settings` and get its own admin page rather than living on this one.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_server_setup.cfm`	`hermes_commandbox`	Page
`config/hermes/var/www/html/admin/2/inc/save_server_identity.cfm`	`hermes_commandbox`	Save handler
`config/hermes/var/www/html/admin/2/inc/generate_postfix_configuration.cfm`	`hermes_commandbox`	`main.cf` regen + `postfix reload`
`config/hermes/var/www/html/admin/2/inc/generate_nextcloud_configuration.cfm`	`hermes_commandbox`	NC `config.php` regen (trusted_domains)
`config/postfix-dkim/etc/postfix/main.cf`	`hermes_postfix_dkim` (mounted)	Live Postfix config — regen target
`config/postfix-dkim/etc/postfix/main.cf.HERMES`	`hermes_postfix_dkim` (mounted)	Write-time backup of the previous live config
`/var/www/html/config/config.php` inside `hermes_nextcloud`	`hermes_nextcloud`	Live Nextcloud config — regen target
`parameters` rows where `module = 'postfix'`, `parent_name IN ('myorigin','myhostname')`	`hermes_db_server` (`hermes` DB)	The directive values
`parameters2` row where `parameter = 'server_ip'`	`hermes_db_server` (`hermes` DB)	Host IP

The Postfix reload uses the standard `docker exec hermes_postfix_dkim /usr/sbin/postfix reload` pattern. The Nextcloud regen rewrites the bind-mounted `config.php` directly, no `occ` calls — NC picks up the change on the next request because `config.php` is read per-request.

Related

- Console Settings — the web-side identity (Console Address, Console Certificate). Companion to this page.
- SMTP TLS Settings — bind a TLS certificate to the Mail Server Hostname so STARTTLS handshakes verify
- System Certificates — issue / renew the cert that SMTP TLS Settings binds
- System Settings — other globals (timezone, language) not part of server identity
- Release engineering and updates — initial install flow that populates these values for the first time

---

Revision #14
Created 2026-05-31 12:52:02 UTC by Dino Edwards
Updated 2026-06-13 12:30:06 UTC by Dino Edwards