

# Administrator Guide

This guide covers everything a **Global Admin** or **Tenant Admin** needs to run Synapse-Cortex: initial setup, user management, SLA configuration, MFA enforcement, the NetscanXi asset-import integration, and the audit trail.

## 1. First-Run Setup

The first time Synapse-Cortex starts against an empty database, every route redirects to `/setup` until an admin account exists. There is no default login — you create it yourself.

1. Run `docker compose up --build` from the project root (see the top-level README/`.env.example` for `POSTGRES_*` and `SECRET_KEY` values).
2. Visit `http://<host>:8000/setup`.
3. Enter your **Tenant Name**, plus your own name, email, and password. This creates the tenant, seeds default SLA policies for all four priorities, and creates you as the tenant's **Global Admin**.
4. You're immediately logged in and land on the Dashboard.

There is one tenant per Synapse-Cortex deployment in this release; all users, tickets, assets, and SLA policies belong to it.

## 2. Roles

Four roles, from most to least privileged:

| Role                      | Can do                                                              |
|---------------------------|---------------------------------------------------------------------|
| <code>global_admin</code> | Everything, including managing other global admins                  |
| <code>tenant_admin</code> | Everything except manage global admin accounts                      |
| <code>agent</code>        | Work tickets: view/update status, priority, assignment, description |
| <code>requester</code>    | Create tickets, view their own                                      |

Only `global_admin` and `tenant_admin` see the **Admin** menu (Users, SLA Policies, Asset Types, Integrations, Audit Trail). A `tenant_admin` cannot create, edit, or delete a `global_admin` account — only another `global_admin` can.

Deleting a ticket is also admin-only: the **Delete Ticket** button on a ticket's detail page only appears for `global_admin` and `tenant_admin`. Deletion is permanent (it also removes that ticket's logged actions) and is recorded in the Audit Trail.

### 3. Managing Users

**Admin** → **Users** lists every account in the tenant.

- **Create:** click + *New User*, fill in name/email/password/role. Passwords must be at least 8 characters.
- **Change role:** use the inline role dropdown on any row (subject to the role-hierarchy rule above).
- **Enable/disable:** click the status badge to toggle an account on or off without deleting it.
- **Delete:** click *Delete* on any row except your own — you can't delete yourself.
- **Require MFA:** click the MFA badge to toggle "required" for that user. See below for what happens next.

### 4. Multi-Factor Authentication (MFA)

MFA is TOTP-based (any standard authenticator app — Google Authenticator, Authy, 1Password, etc.).

- **Self-service enrollment:** any user can go to **Settings** → **Profile & MFA** and click *Enable MFA* to scan a QR code and confirm with a 6-digit code.
- **Admin-required MFA:** if you flag a user as "MFA required" from the Users page and they haven't enrolled yet, their **next login redirects straight to the enrollment screen** — they can't reach the dashboard until they finish setting it up.
- **Disabling:** a user can turn MFA off themselves from Settings by re-entering their password. There is no tenant-wide "force MFA for everyone" switch in this release — it's set per user.

### 5. SLA Policies

**Admin** → **SLA Policies** shows the four priority tiers (Critical/High/Medium/Low) with their response and resolution time targets in minutes. Defaults are seeded at setup; edit the numbers inline and click *Save* per row. Changes apply to tickets created afterward — existing tickets keep the due dates computed under the old policy.

### 6. Managing Asset Types

**Admin** → **Asset Types** controls the list of categories available in the asset **Type** field, both for assets created manually and for assets imported from NetscanXi. Every new tenant starts with seven defaults (Server,

Workstation, Laptop, Network Device, Mobile, Printer, Other), but these are just a starting point — rename, add, or remove them to match how your organization actually categorizes hardware.

- **Add a type:** click + *New Type* and give it a name (unique within the tenant).
- **Rename a type:** edit the name inline in the table and click away (or tab out) to save. Any asset already using that type is unaffected — it keeps pointing at the same category, now shown under its new name.
- **Delete a type:** click *Delete*. If any asset currently uses that type, deletion is blocked — reassign those assets to a different type first.

Renaming or adding types also changes what NetscanXi's imports match against: on each import, NetscanXi's free-text device-type guess is matched against your tenant's *current* type names (see Section 8), so keeping familiar defaults like "Server" or "Printer" around helps that matching stay accurate.

## 7. Audit Trail

**Admin** → **Audit Trail** is a running log of every meaningful action in the tenant: ticket and asset creates/updates/deletes, ticket actions logged, asset type changes, asset imports, user management, SLA policy changes, logins (success and failure), MFA enrollment/disable, and API key generation/revocation. Filter by action type with the dropdown; results are paginated 25 at a time. Asset imports from NetscanXi log **one row per push batch** (not per asset), so a 500-device scan doesn't flood the log — the row's detail shows the created/updated counts for that batch.

## 8. NetscanXi Integration (Assets + Tickets)

This is the integration that keeps your CMDB and ticket queue in sync with what NetscanXi actually finds on the network. One API key covers two feeds: **asset inventory** (asset ID, MAC, IP, device type, OS, software) and **tickets** (auto-created from NetscanXi's Remediation Tracking items).

### How it works

NetscanXi **pushes** to Cortex; Cortex never reaches out to NetscanXi. The asset ID NetscanXi assigns to a device is used **directly** as that asset's ID in Cortex too, so the same device shows the same ID in both tools. (Assets you create manually in Cortex keep the existing `AST-xxxxxxxxxx` ID scheme; only NetscanXi-imported assets get NetscanXi's raw ID.)

### Setup steps

1. In Cortex, go to **Admin** → **Integrations**.
2. Click **Generate API Key**. The key is shown exactly once — copy it now along with the **API URL** shown above it (this is the *base* address of the Cortex instance, with no path after it — NetscanXi appends its own ingest paths automatically). If you lose the key, you'll need to generate a new one (the old one stops working immediately).
3. In NetscanXi, open the **Synapse Cortex** panel (toolbar, tenant-admin only).

4. Paste the API URL and API Key, check **Enable sending asset data to Synapse Cortex**, and click **Save**. This one toggle and key pair governs both feeds below.
5. Click **Test Connection** — you should see "OK Connected". If you see "Failed to Connect", double-check the URL and key (and that Cortex is reachable from NetscanXi's network).
6. Click **Send latest scan now** to push the asset inventory, and/or **Send remediation items as tickets** (further down the same panel) to push tickets. Either can be triggered independently, whenever you like.

## What gets imported — Assets

Each pushed asset creates or updates a Cortex Asset record:

- **Asset ID** — NetscanXi's ID, used as-is.
- **MAC / IP** — as discovered. Devices without a MAC (e.g. some Wi-Fi-only or virtual hosts) are still imported; Cortex only enforces "no duplicate MAC per tenant" for assets that actually have one.
- **Asset Type** — NetscanXi's free-text device-type guess is matched on a best-effort basis against your tenant's *current* list of asset types (see Section 6), not a fixed set — so if you've renamed or removed a default category, the match adapts accordingly, falling back to "Other" (or whatever type is available) when nothing matches. The original NetscanXi label is always preserved and shown on the asset's detail page as "as reported by NetscanXi," even if the match guessed wrong.
- **OS and Software** — shown on the asset's detail page under **Discovered OS & Software**, and can be corrected there if needed.

Re-pushing the same scan updates existing asset records in place — it never creates duplicates, matching first on NetscanXi's asset ID and falling back to MAC address. It's also change-aware: if a re-push finds a device whose data hasn't actually changed since last time, that asset is left completely untouched (no write, no "last updated" bump) rather than being marked updated just because a push happened — only assets that are genuinely new or have genuinely changed data count toward the created/updated totals shown in the Audit Trail.

## What gets imported — Tickets

Each pushed item from NetscanXi's **Remediation Tracking** module creates a Cortex ticket the first time it's seen:

- **Title / Description** — taken from the remediation item; the CVE (if any) is appended to the title.
- **Priority** — passed straight through (NetscanXi's low/high/medium/critical scale matches Cortex's exactly).
- **Status** — mapped from NetscanXi's remediation status: `open`→New, `in_progress`→In Progress, `blocked`→Awaiting User, `resolved`→Resolved.
- **Asset link** — if the remediation item names an asset NetscanXi has already imported, the ticket is linked to it automatically.

Tickets are matched by a stable reference id (`netscanxi:<tenant>:remediation:<id>`) shown on the ticket's detail page as **External Ref**. **Unlike assets, ticket import is create-only**: once a ticket exists for a given remediation item, later pushes leave it alone entirely — title, status, priority, whatever an agent has since changed in Cortex is never overwritten. If NetscanXi later marks the same item resolved, that does not retroactively update the Cortex ticket; close it out in whichever system is doing the work.

## Revoking access

Back in **Admin** → **Integrations**, click **Revoke** to invalidate the key immediately (NetscanXi's next push, of either kind, will get a 401) or uncheck **Enable** in NetscanXi to pause pushing without touching the key.

## 9. Deployment Notes

- Schema changes ship as Alembic migrations, applied automatically on container startup (`alembic upgrade head`). There is no manual migration step for a standard Docker Compose deployment.
- The app requires Postgres in production; the `DATABASE_URL` in `docker-compose.yml` points at the bundled `db` service by default.