

# Recurring Checklist

Self-hosted deployment guide — Synology + Docker + Gmail SMTP

Document version 1 · Generated for IT handoff · *Edit any value in this document to match your environment.*

## Contents

1. What this is & what was deployed
2. Network architecture
3. Deployment steps (already completed)
4. How users access it
5. How admin manages it
6. Weekly email digest
7. Backups (Hyper Backup → Google Drive)
8. Updating the app code
9. Troubleshooting / common ops
10. Credentials & references

---

## 1. What this is

A small Node.js web application running in a Docker container on the Synology NAS named **Chronicles-NA**. It hosts a multi-location checklist that staff complete each Sunday for startup and shutdown procedures. Every Sunday at 3:00 PM the server emails a single digest to [media@waterschurch.org](mailto:media@waterschurch.org) summarising what each location checked off and what they missed, then resets the list for the next week.

## Components

| Piece                       | Where                                                                     | Purpose                                                                                           |
|-----------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Web app (Node.js + Express) | Container on Chronicles-NA, port 3000                                     | Serves the UI and stores check states                                                             |
| JSON state file             | <code>/volume1/docker/recurring-checklist/data/checklists.json</code>     | Source of truth for lists, items, and per-location checks                                         |
| Reverse proxy               | Boba-Fett (DSM → Control Panel → Login Portal → Advanced → Reverse Proxy) | Terminates HTTPS for the friendly URL and forwards to Chronicles-NA                               |
| Let's Encrypt certificate   | Boba-Fett (DSM → Control Panel → Security → Certificate)                  | Issued for <code>productionchecklist.synology.me</code> ; auto-renews every 90 days               |
| Scheduler (node-cron)       | Inside the container                                                      | Fires <code>0 15 * * 0</code> in <code>America/New_York</code> to send the digest and reset state |
| Hyper Backup task           | Chronicles-NA → Google Drive                                              | Daily 23:30 backup of the <code>docker</code> shared folder                                       |

## 2. Network architecture



### Hosts & URLs

| Hostname                                             | Role                                                  |
|------------------------------------------------------|-------------------------------------------------------|
| <code>https://productionchecklist.synology.me</code> | Public URL for the app (TLS via Boba-Fett)            |
| <code>http://10.1.1.1:3000</code>                    | Direct LAN access to the app (bypasses reverse proxy) |
| <code>chronicles-na</code> (10.1.1.1)                | Synology that runs the container                      |
| <code>boba-fett</code> (10.1.1.4)                    | Synology that owns the public IP & reverse proxy      |

### 3. Deployment steps (already completed)

The system is already deployed. The steps below document exactly how it was built so it can be reproduced on another NAS if needed.

#### 3.1 Prepare the source bundle

The application source lives in this repository under `server/`. A deploy tarball is produced from that folder (excluding `node_modules`):

```
tar --exclude='node_modules' --exclude='.DS_Store' \
  -czf checklist-deploy.tgz -C server .
```

#### 3.2 Upload to the NAS

1. Open **File Station** on Chronicles-NA.
2. Navigate to the `docker` shared folder (created automatically when Container Manager is installed).
3. Create folder `recurring-checklist` inside `docker`.
4. Upload `checklist-deploy.tgz` into the new folder.
5. Right-click → **Extract** → Extract here. Synology unpacks `.tgz` in two passes (it leaves an intermediate `.tar`) — right-click that `.tar` and extract again.
6. Delete `checklist-deploy.tgz`, `checklist-deploy.tar`, and the `PaxHeader` folder.

#### 3.3 Configure `.env`

The bundle ships a working `.env`. The values that matter are listed below. Edit through File Station → right-click → **Open with** → **Text Editor** (or any text editor).

| Key                | Value used                                                                      | Notes                                                                                                                        |
|--------------------|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| PORT               | 3000                                                                            | Container-internal port. Don't change unless also editing <code>docker-compose.yml</code> .                                  |
| ADMIN_PASSWORD     | (set in <code>.env</code> )                                                     | Password used to enter "Admin" mode in the web UI.                                                                           |
| REPORT_EMAIL       | <code>media@waterschurch.org</code>                                             | Recipient of the Sunday digest.                                                                                              |
| REPORT_CRON        | <code>0 15 * * 0</code>                                                         | Sunday at 3:00 PM in <code>REPORT_TZ</code> .                                                                                |
| REPORT_TZ          | <code>America/New_York</code>                                                   | IANA timezone for the cron.                                                                                                  |
| RESET_AFTER_DIGEST | <code>true</code>                                                               | Server clears all check states after the digest is sent.                                                                     |
| SMTP_HOST          | <code>smtp.gmail.com</code>                                                     |                                                                                                                              |
| SMTP_PORT          | <code>587</code>                                                                | STARTTLS port.                                                                                                               |
| SMTP_USER          | <code>media@waterschurch.org</code>                                             | Sender — a Google Workspace account.                                                                                         |
| SMTP_PASS          | <i>Gmail App Password (16 chars)</i>                                            | Generated at <code>myaccount.google.com/apppasswords</code> . NOT the regular Gmail password — requires 2-Step Verification. |
| SMTP_FROM          | <code>Recurring Checklist</code><br><code>&lt;media@waterschurch.org&gt;</code> |                                                                                                                              |

### 3.4 Build and start the container

1. Install **Container Manager** from Package Center on Chronicles-NA (creates the `docker` shared folder automatically).
2. Open Container Manager → left sidebar **Project** → **Create**.
3. Project name: `recurring-checklist`. Path: `/volume1/docker/recurring-checklist`. Source: "Use existing docker-compose.yml".
4. Skip the "Set up web portal via Web Station" option (not needed).
5. Click Done. Container Manager runs `docker compose up`; first build takes ~60 seconds. Final log line should be `Exit Code: 0`.
6. Verify: `http://10.1.1.1:3000` serves the app.

### 3.5 Public hostname + TLS

1. **DDNS** (Chronicles-NA → Control Panel → External Access → DDNS → Add): Synology provider, hostname `productionchecklist.synology.me`, External Address Auto. Status should be Normal.
2. **Reverse proxy** on *Boba-Fett* (Control Panel → Login Portal → Advanced → Reverse Proxy → Create):
  - Source: `HTTPS` · `productionchecklist.synology.me` · port `443`
  - Destination: `HTTP` · `10.1.1.1` · port `3000`

3. **Let's Encrypt** on Boba-Fett (Control Panel → Security → Certificate → Add → Get a certificate from Let's Encrypt):

- Domain: `productionchecklist.synology.me`
- Email: `media@waterschurch.org`

4. Bind the cert: **Settings** button in the Certificate tab → set the `productionchecklist.synology.me` row to use the new cert → OK.

#### Why does the reverse proxy live on Boba-Fett, not on Chronicles-NA?

The router forwards port 443 to Boba-Fett. Putting the proxy there avoids any router changes. Boba-Fett receives `productionchecklist.synology.me` traffic, terminates TLS with the Let's Encrypt cert, and forwards plain HTTP to Chronicles-NA:3000 over the LAN.

## 4. How users access it

Any browser on any device:

```
https://productionchecklist.synology.me
```

First visit on each device asks for a name. That name is the **location identifier** (e.g. `Woonsocket`, `North Attleboro`). Case and whitespace are normalised — `woonsocket`, `WOONSOCKET`, and `Woonsocket` all collapse to the same canonical name. The name is stored in the browser's `localStorage` and can be changed by clicking the name pill in the top-right.

Each location ticks off items independently. Unassigned lists are visible to all locations but checks are per-location. Lists with a specific location are only visible to (and writeable by) that location.

## 5. How admin manages it

1. Click the  **Admin** button in the top-right and enter `ADMIN_PASSWORD`.
2. A horizontal "Menu" bar appears under the list tabs. Click or swipe it to expand:
  - **+ Add List / Rename List / Delete List** — each list has an optional "Assign to location" field. Leave blank to show the list to every location.
  - **Export JSON** — downloads the current state file as a backup snapshot.
  - **Import JSON** — appends imported lists into the current state, auto-tagged with the current logged-in location.
  - **Send Digest Now** — sends a one-off (manual) copy of the Sunday report immediately. Useful to verify SMTP. Does *not* reset state.
3. Within a list (admin mode), each item has Edit / Move / Delete buttons. Drag tabs sideways to reorder lists; drag items up/down within the To Do or Done column to reorder

tasks.

4. Click the yellow  **Admin (on)** button again to leave admin mode. All management UI disappears for non-admins.

---

## 6. Weekly email digest

---

A `node-cron` job fires at `0 15 * * 0` in `America/New_York` (Sunday 3:00 PM local). It:

1. Reads the current state from `checklists.json`.
2. Builds a per-location report: one section per location, each list within shown as a green/red table of ✓/X items with the who+when stamp.
3. Sends the email via `smtp.gmail.com:587` using the configured app password, to `REPORT_EMAIL` only. The subject is `Checklist Report – <weekday, month, day>`.
4. If `RESET_AFTER_DIGEST=true`, the entire check-states map is cleared so the next week starts clean. Items themselves are preserved.

To change the schedule, edit `REPORT_CRON` in `.env` and rebuild the container (see §8).

---

## 7. Backups

---

Hyper Backup task “**Recurring Checklist**” on Chronicles-NA:

- Destination: Google Drive (signed in as `media@waterschurch.org`).
- Source: `docker` shared folder (includes everything under `/volume1/docker`).
- Schedule: Daily 23:30; integrity check Saturday 01:30.
- Compression on. Client-side encryption off.

Restoring an entire run is as simple as: open Hyper Backup → select task → Restore Data → pick a version → restore to `/volume1/docker/recurring-checklist/data/checklists.json`. The container picks up the new file on next request (no restart needed).

---

## 8. Updating the app code

---

The state file ( `data/checklists.json` ) and the configuration file ( `.env` ) are preserved between updates because they live in a Docker bind-mount and on disk. The update procedure is:

1. Developer produces a new `checklist-deploy.tgz` from the latest `server/` source.
2. Upload the new tarball into `/volume1/docker/recurring-checklist/` via File Station.
3. **Important:** before extracting, rename or move the existing `.env` aside so it's not overwritten (the bundle ships an example `.env`, but if the deploy script excludes it, this step is unnecessary — confirm with the developer per release).

4. Extract the new tarball (in two passes, as in §3.2). Replace existing files when prompted.
5. Delete the leftover `.tgz` / `.tar` / `PaxHeader` .
6. Open Container Manager → Project → **recurring-checklist** → **Action** menu → **Build** (or Stop, then Start with Build). This rebuilds the image with the new code and restarts the container.
7. Verify: `http://10.1.1.1:3000` still loads, and any new functionality from the release is visible.

Rollback: keep the previous `checklist-deploy.tgz` archived. To roll back, re-extract the older bundle, rebuild. State file is untouched.

## 9. Troubleshooting / common ops

| Symptom                                                | Where to look                                                       | Fix                                                                                                                                                                                                                       |
|--------------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App URL times out                                      | Container Manager → recurring-checklist → status                    | If stopped: Action → Start. If crashing: open container Log, look for errors.                                                                                                                                             |
| Digest email never arrived                             | Container Manager → container Log, search <code>[digest]</code>     | Look for <code>Failed: ...</code> . Most often Gmail app password revoked or rotated; regenerate at <code>myaccount.google.com/apppasswords</code> and update <code>SMTP_PASS</code> in <code>.env</code> , then rebuild. |
| Cert renewal failed                                    | Boba-Fett → Control Panel → Security → Certificate                  | Right-click the cert → Renew. Requires port 80 reachable on Boba-Fett.                                                                                                                                                    |
| State got corrupted / accidental wipe                  | Hyper Backup → Recurring Checklist task                             | Restore Data → pick latest known-good version → restore the <code>data/</code> folder only.                                                                                                                               |
| “Wrong password” on admin login                        | <code>.env</code> on the NAS                                        | Compare <code>ADMIN_PASSWORD</code> . If changed, rebuild container (env vars are read on container start).                                                                                                               |
| Reverse proxy returns Boba-Fett DSM instead of the app | Boba-Fett → Control Panel → Login Portal → Advanced → Reverse Proxy | Confirm rule for <code>productionchecklist.synology.me</code> exists and routes to <code>10.1.1.1:3000</code> . Cert must be bound (see §3.5).                                                                            |

### Useful one-liners (SSH into Chronicles-NA)

```
# See the running container
sudo docker ps | grep recurring

# Tail logs
sudo docker logs -f recurring-checklist

# Manually restart the container
sudo docker compose -f /volume1/docker/recurring-checklist/docker-compose.yml

# Look at the live state file
cat /volume1/docker/recurring-checklist/data/checklists.json | head
```

---

## 10. Credentials & references

---

- **Admin password:** set in `.env` on the NAS (key `ADMIN_PASSWORD`). Rotate by editing `.env` and rebuilding the container.
- **Gmail app password:** generated at <https://myaccount.google.com/apppasswords> using the `media@waterschurch.org` Google Workspace account. Stored in `.env` as `SMTP_PASS`. To rotate: revoke the existing app password in Google, generate a new one, paste into `.env`, rebuild.
- **Synology DDNS:** `productionchecklist.synology.me` tied to the Synology Account used during DDNS setup. Renewal is automatic.
- **Let's Encrypt cert:** 90-day validity, auto-renewal handled by DSM 7. Confirm under Control Panel → Security → Certificate.
- **Source repository:** the application source is the `server/` directory of `recurring-checklist-app`, maintained by the application developer.

### Security notes

The app uses a shared password for admin actions (no per-user accounts). Anyone on the open URL can check items off using any name, so don't treat the contents of the checklist as confidential. The admin password is the only gate for adding/editing/deleting lists.