

Recurring Checklist

Deployment & maintenance guide — self-hosted web app on Synology DSM.

Contents

1. Overview
2. Environment summary
3. Initial deployment
4. Pushing app updates
5. Backups
6. DDNS, reverse proxy & HTTPS
7. Troubleshooting
8. Reference

1. Overview

The Recurring Checklist is a small Node.js web application that lives in a Docker container on the Synology NAS. Multi-location operators (Woonsocket, North Attleboro, etc.) open it in a web browser, identify themselves once per device, and tick items off through the week. Every Sunday at 3:00 PM the server emails a digest of who completed what (and what was missed) to a single recipient, then resets all checkboxes for the following week.

ARCHITECTURE

Users (LAN or internet)



https://productionchecklist.synology.me (Boba-Fett DSM, port 443)



reverse proxy



http://10.1.1.1:3000 (Chronicles-NA DSM, Docker container)



node:20-alpine + Express + node-cron + nodemailer



└─ data volume → /volume1/docker/recurring-checklist/data/checklists.json



Backup)

(backed up nightly to Google Drive via Hyper



└─ Sunday 3:00 PM cron → Gmail SMTP → media@waterschurch.org

2. Environment summary

Item	Value
App URL (external)	<code>https://productionchecklist.synology.me</code>
App URL (LAN, direct)	<code>http://10.1.1.1:3000</code>
Public TLS host	Boba-Fett (10.1.1.4), DSM reverse proxy + Let's Encrypt cert
Container host	Chronicles-NA (10.1.1.1), Container Manager project <code>recurring-checklist</code>
Project folder on NAS	<code>/volume1/docker/recurring-checklist/</code>
Persisted data file	<code>/volume1/docker/recurring-checklist/data/checklists.json</code>
Backups	Hyper Backup → Google Drive (<code>recurring-checklist_server-backup/Chronicles-NA_1</code>), daily 23:30
Email digest schedule	Sunday 15:00 America/New_York
Email recipient	<code>media@waterschurch.org</code>
SMTP relay	<code>smtp.gmail.com:587</code> (Google Workspace app password)
Admin password	Set in <code>.env</code> as <code>ADMIN_PASSWORD</code>

3. Initial deployment

The container was created using the following procedure. Use this section as the disaster-recovery / rebuild reference.

3.1 Prerequisites

- Synology DSM 7.2 or newer on the host NAS (Chronicles-NA)

- Container Manager package installed
- SSH or File Station access to `/volume1/docker/`
- Google Workspace account with an App Password (for the digest sender)
- The deployment tarball: `checklist-deploy.tgz` (produced from the project's `server/` directory)

3.2 Upload the bundle

1. Open File Station on the NAS.
2. Inside `/volume1/docker/`, create a folder named `recurring-checklist`.
3. Upload `checklist-deploy.tgz` into it.
4. Right-click → Extract → Extract here. Repeat once on the resulting `.tar` (Synology's File Station unwraps in two passes).
5. Delete the `.tgz`, `.tar`, and the `PaxHeader` folder that gets produced. The folder should be left with:

```
server.js      public/      Dockerfile
package.json   data/        docker-compose.yml
package-lock.json README.md    .env .env.example
.dockerignore  .gitignore
```

3.3 Configure environment

The deployment ships with a working `.env` file. Verify the following keys are populated:

Variable	Purpose
ADMIN_PASSWORD	Required for any list-management operation in the UI.
REPORT_EMAIL	Where the Sunday digest is delivered.
REPORT_CRON	cron schedule for the digest. Default <code>0 15 * * 0</code> (Sun 3:00 PM).
REPORT_TZ	IANA timezone for the cron (default <code>America/New_York</code>).
RESET_AFTER_DIGEST	<code>true</code> = clear all checks once the digest is sent. <code>false</code> to disable.
SMTP_HOST	<code>smtp.gmail.com</code>
SMTP_PORT	<code>587</code>
SMTP_USER	Gmail / Google Workspace address that sends the digest.
SMTP_PASS	16-character app password from <code>myaccount.google.com/apppasswords</code> .
SMTP_FROM	Friendly <code>From:</code> header, e.g. <code>Recurring Checklist <media@waterschurch.org></code>

Security note

The `.env` file contains credentials. Do not place it in a publicly-shared folder. Hyper Backup is encrypted in transit; client-side encryption is not enabled because it complicates restore.

3.4 Create the Container Manager project

1. Open Container Manager → **Project** → **Create**.
2. **Project name:** `recurring-checklist`
3. **Path:** browse to `/volume1/docker/recurring-checklist` → Select.
4. **Source:** "Use existing docker-compose.yml" (auto-selected). Click **OK**.
5. On the *Web portal settings* screen, leave the checkbox unchecked. Click **Next**.
6. Confirm the summary and click **Done**.

Container Manager will build the image (~60–90 seconds on first run) and start the container. The project should report `Exit Code: 0` at the end of the build log and the project status indicator should turn green.

3.5 Verify

1. From a browser on the LAN, open `http://10.1.1.1:3000`. The "Who are you?" prompt should appear.
2. Click **Admin** (top right) and authenticate with the value of `ADMIN_PASSWORD`.
3. Use **Send Digest Now** in the admin drawer to verify SMTP connectivity. A test email should arrive within a few seconds.

4. Pushing app updates

Updates are delivered as a refreshed deployment tarball. Application data is mounted from the host's `data/` directory, so rebuilding the image preserves all checklists, locations, and check states.

4.1 Receive the update

The developer produces a fresh `checklist-deploy.tgz` from the latest project tree. It will arrive as a single file (currently ~35 KB).

4.2 Apply on the NAS

1. Open File Station → navigate to `/volume1/docker/recurring-checklist/`.
2. Upload the new `checklist-deploy.tgz` into this folder.
3. Right-click → Extract → Extract here. Choose **Overwrite** when prompted (this replaces `server.js`, the `public/` folder, etc.; it does *not* touch `data/`).
4. Repeat the extract on the resulting `.tar`.
5. Delete the `.tgz`, `.tar`, and `PaxHeader` when done.

4.3 Rebuild the container

1. Open Container Manager → **Project** → click the `recurring-checklist` row.
2. Click **Action** → **Build** (rebuilds the image from the updated source).
3. Wait for the build log to finish with `Exit Code: 0`.
4. Click **Action** → **Stop**, then **Action** → **Start** if the build did not auto-restart.

What gets preserved across updates

Everything in `./data/` is volume-mounted. Lists, items, per-location check states, and `lastCheckDate` values are persisted. Configuration in `.env` is also untouched by an update unless the new tarball explicitly ships a replacement.

4.4 Rollback

If an update misbehaves:

1. Recover the previous `checklist-deploy.tgz` (kept by the developer) or restore from Hyper Backup.
2. Replace the files in `/volume1/docker/recurring-checklist/` as in §4.2.
3. Rebuild the project as in §4.3.

5. Backups

5.1 What's backed up

A Hyper Backup task named **Recurring Checklist** backs up the entire `docker` shared folder nightly at 23:30 to Google Drive. Destination:

```
Google Drive: recurring-checklist_server-backup / Chronicles-NA_1
```

This captures both the application source (in `recurring-checklist/`) and its data folder.

5.2 Restoring

To restore the data file only:

1. Open Hyper Backup → Recurring Checklist task → **Restore** → **Data**.
2. Pick a version. Browse into `recurring-checklist/data/` and restore `checklists.json`.
3. Container Manager → Project → Stop, then Start the `recurring-checklist` project to pick up the restored file.

5.3 Backup verification

An integrity check runs weekly on Saturday at 01:30. Failed integrity checks generate a DSM notification.

6. DDNS, reverse proxy & HTTPS

6.1 DDNS record

- **Hostname:** `productionchecklist.synology.me`
- **Provider:** Synology (free)
- **Account:** Registered to the Synology account on Chronicles-NA.

6.2 Reverse proxy rule (on Boba-Fett, 10.1.1.4)

Located at *Control Panel* → *Login Portal* → *Advanced* → *Reverse Proxy* → "Recurring Checklist".

Side	Protocol	Hostname	Port
Source	HTTPS	<code>productionchecklist.synology.me</code>	443
Destination	HTTP	<code>10.1.1.1</code>	3000

6.3 TLS certificate

- **Type:** Let's Encrypt
- **Bound to:** the `productionchecklist.synology.me` service in Control Panel → Security → Certificate → Settings
- **Auto-renewal:** handled by DSM. The cert renews ~30 days before expiry as long as ports 80/443 inbound continue to reach Boba-Fett.

6.4 Router (UniFi)

Existing port-forward of TCP 443 to Boba-Fett (10.1.1.4) is the only inbound rule required. No new forwarding is needed for the Recurring Checklist app.

7. Troubleshooting

Symptom	Likely cause & fix
App returns 502 / Bad Gateway from <code>productionchecklist.synology.me</code>	The Docker container on Chronicles-NA is stopped. Open Container Manager → Project → click <code>recurring-checklist</code> → Action → Start . If it fails to start, open the build log for errors (commonly a malformed <code>.env</code>).
Browser shows "Not Secure" on the public URL	Certificate either expired or not bound. Check Control Panel → Security → Certificate on Boba-Fett. The <code>productionchecklist.synology.me</code> cert should be green and bound under Settings .
Sunday email did not arrive	Open Container Manager → Container → <code>recurring-checklist</code> → Details → Log . Look for lines beginning with <code>[digest]</code> . Common causes: expired Google App Password, Google Workspace 2FA disabled on the sending account, outbound TCP 587 blocked by firewall.
"This list is reserved for X" error	Expected. The list is location-scoped; a user signed in as a different location cannot tick it off. Either change the user's location to match the list or remove the list's location to make it shared.
Checks reappear after a user signs out and back in	Per-location checks are stored on the server keyed by the user's typed name (case-insensitive, whitespace-collapsed). If a user enters a different name, they are treated as a different location and see different state.
Container won't build	Usually a stale or partial extraction. Re-upload <code>checklist-deploy.tgz</code> and extract both <code>.tgz</code> and the intermediate <code>.tar</code> step. Confirm <code>server.js</code> and the

`public/` folder are present at the project root.

8. Reference

8.1 Files in `/volume1/docker/recurring-checklist/`

Path	Purpose
<code>server.js</code>	Express app + scheduler + email logic.
<code>public/</code>	Static front-end (HTML, CSS, JS).
<code>Dockerfile</code>	Image definition (node:20-alpine).
<code>docker-compose.yml</code>	Container Manager project definition.
<code>package.json</code> / <code>package-lock.json</code>	npm manifest, locked dependency versions.
<code>.env</code>	Runtime config (passwords, SMTP, schedule). Not in version control.
<code>.env.example</code>	Template for a fresh deploy.
<code>data/checklists.json</code>	Persistent state. Volume-mounted; backed up nightly.

8.2 Useful container actions

Action	Where
View live logs	Container Manager → Container → <code>recurring-checklist</code> → Details → Log
Restart container	Container Manager → Project → recurring-checklist → Action → Stop, then Start
Rebuild image after update	Container Manager → Project → recurring-checklist → Action → Build
Edit env vars	Edit <code>/volume1/docker/recurring-checklist/.env</code> via File Station, then restart the project

8.3 Server endpoints (for diagnostics)

Method	Path	Purpose
GET	<code>/api/state</code>	Returns current lists, items, and per-location check map (read-only).
POST	<code>/api/admin/verify</code>	Validates the admin password (used by the UI).
POST	<code>/api/admin/state</code>	Replaces full state. Requires <code>X-Admin-Password</code> .
POST	<code>/api/admin/send-digest</code>	Manually triggers the email digest. Requires admin.
POST	<code>/api/check</code>	Toggle a single item's checked state for a location.

Document version 1 · Generated for the Recurring Checklist deployment on Chronicles-NA / Boba-Fett. Maintained alongside the project source.