

UF-Xray Deployment Guide

Version: 1.0 Date: 2025-09-20

1) Repo Layout (deploy-related)

- **Frontend:** React at repo root. Workflow: `.github/workflows/deploy.yml`.
- **Backend:** `server/` (Express + Python). Dockerfile inside.
- **Render** blueprint: `render.yaml` at repo root.

2) Frontend — GitHub Pages (CI/CD)

- Package homepage: `"homepage": "https://<username>.github.io/UF_Xray"`
- Workflow steps: `checkout` → `Node 18` → `npm ci` → `build` → `upload` → `deploy Pages`.
- Build env: `REACT_APP_API_URL` from GitHub Secrets, `CI=false` to ignore warnings.
- Trigger: push to `main` or manual Run workflow.
- Routing: HashRouter supports `/#/News` etc. on static hosting.

3) Backend — Render (Docker)

- Dockerfile installs Node + Python + pip dependencies, exposes `5000`, runs `npm start`.
- `render.yaml` defines a web service with `rootDir: server` and `healthCheckPath: /healthz`.
- Deploy via: Render → New → Blueprint → select repo → Create Resources → Deploy.
- After LIVE, copy the service URL (e.g., `https://uf-xray-api.onrender.com`).
- Verify: `/healthz`, `/api/news`, `/api/scan-url`, `/api/scan-file`, `/api/scan-log`.

4) Connect Frontend ↔ Backend

- GitHub Secret: `REACT_APP_API_URL` => your backend URL.
- Re-run Pages workflow so the built app points to the correct API.
- Local dev falls back to `http://localhost:5000` when on localhost.

5) Mixed-Content Safe News Images

- Some feeds serve `http://` images; browsers block them on `https://`.
- Backend proxy GET `/api/news-image?src=...` serves the image over HTTPS.
- Frontend routes `http://` images through the proxy, keeps `https://` images direct.

6) Manual Frontend Deploy (optional)

```
echo REACT_APP_API_URL=https://your-backend > .env.production  
npm install  
npm run deploy
```

7) Troubleshooting

- **Pages 404:** repo Public, Pages Source = GitHub Actions, wait 1–5 min, use `/#/` routes.
- **API fails:** secret set? backend live? check `/healthz`. Re-run workflow after changing secrets.
- **RSS images:** ensure backend with proxy is deployed; check network tab for `/api/news-image`.

8) Maintenance

- Frontend: push to main auto-deploys Pages.
- Backend: Render Blueprint can autodeploy on push.
- Rotate/Update `REACT_APP_API_URL` if backend URL changes.