

Déploiement Unraid

Déploiement Unraid

“ Dernière mise à jour : 2026-05-10

Architecture des containers

```
unraid/  
├─ Container `boardgame-referee` (image gitea.thymon.fr/thymon/boardgame-referee:latest)  
├─ Container `qdrant` (image qdrant/qdrant)  
├─ Container TEI (text-embeddings-inference, GPU RTX 3060)  
└─ Container TEI Reranker (text-embeddings-reranker, GPU RTX 3060)
```

VM externe : `oracle` (Claude Code CLI), accédée via SSH.

docker-compose.yml (prod)

Fichier : `/mnt/user/appdata/boardgame-referee/docker-compose.yml`

Service `app`

```
app:  
  image: gitea.thymon.fr/thymon/boardgame-referee:latest  
  # Pas de port exposé sur l'hôte (reverse proxy NPM)  
  env_file: .env  
  volumes:  
    - /mnt/user/appdata/boardgame-referee/data:/app/data
```

```
- /mnt/user/appdata/boardgame-referee/pdfs:/app/pdfs
- /mnt/user/appdata/boardgame-referee/ssh:/app/ssh:ro
networks:
  - proxy # réseau externe NPM
healthcheck:
  test: ["CMD", "wget", "-q0-", "http://localhost:3000/api/health"]
  interval: 30s
  timeout: 5s
  retries: 3
  start_period: 10s
restart: unless-stopped
```

Service `qdrant`

```
qdrant:
  image: qdrant/qdrant
  networks:
    - proxy
  volumes:
    - /mnt/user/appdata/boardgame-referee/qdrant:/qdrant/storage
  restart: unless-stopped
```

Réseau

`proxy` : réseau Docker externe, partagé avec NPM. Permet à NPM de joindre le container `app` sans exposer son port sur l'hôte.

Volumes (

```
/mnt/user/appdata/boardgame-
referee/ )
```

Sous-dossier	Container	Mount	Contenu
--------------	-----------	-------	---------

data/	app	/app/data	SQLite DB, caches JSON (OCR, contextual, conflicts), data sources cartes, logs
pdfs/	app	/app/pdfs	PDFs uploadés + PNG rendus
ssh/	app	/app/ssh:ro	Clé privée ed25519 oracle (RO)
qdrant/	qdrant	/qdrant/storage	Collections Qdrant

Tous sur le pool ZFS / parity Unraid (selon ta config).

Variables d'env (UI Unraid)

Configurées via le template XML `unraid/boardgame-referee.xml` :

- L'admin Unraid voit chaque var avec son label, type, défaut, et description
- ⚠ **Toute nouvelle var dans `src/config.ts` doit être ajoutée au XML**, sinon l'UI Unraid ne la propose pas

Templates XML inclus

Fichier	Container
<code>unraid/boardgame-referee.xml</code>	App principale
<code>unraid/text-embeddings-inference.xml</code>	TEI bge-m3
<code>unraid/text-embeddings-reranker.xml</code>	TEI reranker

Mise à jour en prod

```
# 1. Pull la nouvelle image (CI a déjà push)
docker compose -f /mnt/user/appdata/boardgame-referee/docker-compose.yml pull app

# 2. Recreate le container avec la nouvelle image
docker compose -f /mnt/user/appdata/boardgame-referee/docker-compose.yml up -d --force-recreate app
```

```
# 3. Tail les logs pour vérifier que ça boot bien
docker logs -f --tail 50 boardgame-referee
```

Ou via l'UI Unraid : Apps → boardgame-referee → Update / Force Update.

Healthcheck

Docker healthcheck : `GET /api/health` toutes les 30s. Si 3 échecs consécutifs → container marqué unhealthy. Tu peux configurer un script Unraid (ou Uptime Kuma) qui restart le container automatiquement sur unhealthy, mais actuellement c'est manuel.

Logs

- **Stdout / stderr Docker** : `docker logs boardgame-referee`
- **Fichier persistant** : `/app/data/logs/server.log` (rotation 50 Mo / 30j via `entrypoint.sh`)

Entrypoint

`entrypoint.sh` (dans l'image Docker) fait :

1. Fix des permissions sur `/app/data` + `/app/pdfs` (gosu PUID:PGID configurable)
2. Rotation log 50 Mo si dépassé
3. Purge archives > 30 jours
4. Lance `node dist/index.js` avec `tee -a /app/data/logs/server.log`

Pas de scaling horizontal

Single instance. SQLite + sessions en mémoire + `setTimeout` cron in-process empêchent le scaling. Si un jour tu veux plus :

- SQLite → Postgres
- Sessions in-memory → table SQL ou Redis
- `setTimeout` → BullMQ ou équivalent

Revision #1

Created 2026-05-10 15:20:02 UTC by thymon

Updated 2026-05-10 15:20:02 UTC by thymon