

Pipeline CI/CD (Gitea)

Pipeline CI/CD (Gitea)

“ Dernière mise à jour : 2026-05-10

Workflow

```
.gitea/workflows/build.yml
```

87 lignes. Trigger : `push` sur `main`.

Job `test`

- **Runner** : `ubuntu-latest`
- **Container** : `node:22-bookworm-slim`
- **Pourquoi container custom** : Alpine (musl libc) ne supporte pas les prebuilts Node fournis par `actions/setup-node` (glibc). `better-sqlite3` n'a pas de prebuilt musl. → Debian Slim + Node 22 + tous les prebuilts.

Étapes

- Checkout code
- Backend : `npm ci → npm run build (tsc) → npm test (vitest --passWithNoTests)`
- Frontend : `npm ci → npm run build (vue-tsc + vite) → npm test`

Si l'un des steps fail, le job suivant (build) ne tourne pas.

Job `build` (depends: test)

- **Runner** : `ubuntu-latest`

Étapes

```
- Setup Docker Buildx
- Login registry:
  server: vars.REGISTRY_URL # gitea.thymon.fr
  username: secrets.REGISTRY_USER
  password: secrets.REGISTRY_PASSWORD
- Metadata: tags `latest` + short SHA (`docker/metadata-action`)
- Build & push:
  - context: .
  - file: ./Dockerfile (multi-stage)
  - push: true
  - tags: gitea.thymon.fr/thymon/boardgame-referee:latest + :sha-XXXXXXX
  - cache-from / cache-to: type=registry (réutilise les couches précédentes)
```

Tags d'image

- `latest` : dernier commit sur main (utilisé en prod pour pull auto)
- `sha-<7chars>` : tag immuable par commit (utilisé pour rollback)

Variables Gitea

Vars (pas secret)

- `REGISTRY_URL` = `gitea.thymon.fr`

Secrets

- `REGISTRY_USER` = compte Gitea avec write sur le package registry
- `REGISTRY_PASSWORD` = token Gitea ou password

Configurés dans Gitea UI : Repo → Settings → Secrets → Actions secrets.

Local : reproduire le build CI

```
# Backend
npm ci
npm run build
npm test

# Frontend
cd frontend
npm ci
npm run build
npm test
```

Si tout passe en local mais foire en CI, c'est probablement un souci d'environnement (Node version, Alpine vs Debian, etc.). Toujours utiliser Node 22 en local.

Déclencheur manuel

Pas configuré actuellement (`workflow_dispatch` absent). Pour relancer un build sans push :

1. Soit faire un commit vide : `git commit --allow-empty -m "chore: rebuild"`
2. Soit re-run le job depuis l'UI Gitea (Actions → workflow run → Re-run jobs)

Latence typique

- Job `test` : ~3-5 min (npm ci + tsc + vitest)
- Job `build` : ~5-10 min (Docker multi-stage build + push)

Total : ~10-15 min par push sur main.

Pas d'environnement de staging

Push direct sur `main` → image `latest` → tu pulls sur Unraid. Pas de branche `develop`, pas de staging URL.

Si tu veux ajouter un staging :

1. Créer une branche `staging`
 2. Étendre `build.yml` avec un trigger `push: branches: [main, staging]`
 3. Tag différencié : `staging-latest` vs `latest`
 4. Container Unraid `boardgame-referee-staging` séparé
-

Revision #1

Created 2026-05-10 15:20:00 UTC by thymon

Updated 2026-05-10 15:20:00 UTC by thymon