

ZAKON — Lockfile Portability in Production Linux Containers

ZAKON — Lockfile Portability in Production Linux Containers

Status: ACTIVE **Created:** 2026-07-28 **Origin:** Bilko MC #9616 / #9619 — 4-iteration CI repair exposed macOS-generated lockfile drift in Linux Docker builds. **Applies to:** Bilko, Drop, Tok, and any ALAI Node workspace deployed in Linux containers.

Rule

Every deployable Node lockfile (`package-lock.json`, `npm-shrinkwrap.json`, `pnpm-lock.yaml`, `yarn.lock`) for Bilko/Drop/Tok MUST be regenerated inside the same Linux base image used by the production Dockerfile before it is committed for a container deploy.

Do not trust a lockfile generated on macOS for a Linux container build. Optional native dependencies are platform-specific.

Required Procedure

1. Identify the production Dockerfile and its Node base image.
2. Run lockfile regeneration inside that exact base image or a materially identical Linux variant.
3. Remove stale host artifacts before install.
4. Commit the regenerated lockfile with evidence that Linux optional packages are present.

Example for Bilko web (`apps/web/Dockerfile` currently uses the Node production build context):

```
docker run --rm -v "$PWD:/work" -w /work node:20-bookworm-slim bash -c \  
  "rm -rf node_modules package-lock.json && npm install --workspaces --legacy-peer-deps"
```

If the Dockerfile base image changes, the command must change with it.

Acceptance Evidence

A task/PR that changes or relies on a Node lockfile for container deploy must include:

- Docker base image used for regeneration.
- Exact regeneration command.
- Clean command exit code.
- Lockfile diff showing Linux platform optional dependencies when applicable.
- Local Docker build success before remote CI dispatch; see `~/system/rules/zakon-local-docker-build.md`.

Blockers

Block the task before push/dispatch if:

- The lockfile was regenerated on macOS only.
- The base image used for regeneration does not match the production Dockerfile.
- The lockfile diff removes Linux optional/native packages without a documented reason.
- The agent cannot state where the production Dockerfile is.

Why

MC #9619 found that darwin-generated lockfiles omitted Linux optional packages such as native watcher/bundler variants. Cloud/CI builds then failed inside Linux containers even though local macOS installs passed.

Related Standards

- `~/system/rules/zakon-ci-stub-types-contract.md`
- `~/system/rules/zakon-local-docker-build.md`
- Bilko `BUILD-BLUEPRINT.md` §11 and §14
- Bilko `DEPLOY-MAP.md` historical GCP gotchas

Revision #1

Created 2026-07-28 18:18:49 UTC by John

Updated 2026-07-28 18:18:49 UTC by John