



Docker Doctor

50 Checks, 11 Categories, Auto-Fix

FEBRUARY 23, 2026

THEDECIPHERIST.COM

TABLE OF CONTENTS

What it does	3
50 checks in 11 categories	3
Smart compose discovery	4
CI integration	4
Programmatic API	4
Install	5

[View as Website](#)

I kept running into the same Docker issues over and over – containers crashing with cryptic exit codes, compose services that can't talk to each other, bloated images, secrets accidentally baked into layers. Every time it meant 20 minutes of Googling the same things.

So I built **dockerdoctor** – an interactive CLI that runs 50 checks across your Dockerfiles, compose files, and running containers, then tells you exactly what's wrong and how to fix it.

WHAT IT DOES

You run `dockerdoctor` and it walks you through a guided scan:

1. Pick a directory (or let it auto-discover Docker projects in subdirectories)
2. Choose which checks to run (or just run them all)
3. Browse results in a split-pane TUI with severity grouping, syntax-highlighted fix instructions, and one-key clipboard copy

The key difference from other linters: **every finding comes with a concrete fix** – either auto-applied with `--fix`, or with paste-ready code you can copy with `c` and drop straight into your file.

50 CHECKS IN 11 CATEGORIES

Static checks (no Docker needed):

- **Dockerfile (9)** – unpinned `:latest` tags, running as root, shell-form CMD, layer ordering that busts cache, `npm install` vs `npm ci`, missing multi-stage builds, `NODE_ENV` trap (setting it before `npm ci` blocks devDependencies), missing `--chown`, Alpine cache cleanup
- **Compose (7)** – missing healthchecks, undefined networks, network mismatch between services that need to talk, static IPs, host bind mounts, default bridge network, Swarm config silently ignored by `docker compose`
- **Secrets (4)** – hardcoded passwords in ENV/ARG, secrets in compose environment blocks, sensitive files (`*.env`, key files) being COPYed into the image

- **Line Endings (3)** – CRLF that breaks shell scripts in Linux containers, missing `.gitattributes`, no `dos2unix` step
- **Dockerignore (2)** – missing `.dockerignore`, common entries like `node_modules` and `.git` not excluded

Runtime checks (needs Docker daemon):

- **Build (4)** – oversized build context, DNS failures during build, insufficient disk space, platform mismatch (ARM vs x86)
- **Startup (4)** – exit code decoding (137 = OOM, 126 = permission denied, 127 = not found), OOM-killed containers, missing env vars, entrypoint not found
- **Network (4)** – containers on different networks, DNS resolution failures, port conflicts, services bound to localhost
- **Performance (4)** – high CPU/memory, bind mount I/O issues, underutilized build cache, missing resource limits
- **Image (4)** – oversized images, inefficient layers, architecture mismatch via emulation, base image bloat
- **Cleanup (5)** – disk usage, dangling images, stopped containers, unused volumes, build cache waste

SMART COMPOSE DISCOVERY

It doesn't just look for `docker-compose.yml`. If your compose file is named something non-standard like `infra.yml` or `stack.yaml`, `dockerdoctor` sniffs the content for a `services:` key and detects it automatically. It skips false positives like CI configs, Kubernetes manifests, and build tool files.

CI INTEGRATION

```
# GitHub Actions
- name: Lint Docker config
  run: npx dockerdoctor --ci --severity error
```

Exit 0 = clean, exit 1 = errors found. Warnings and info don't fail the build.

PROGRAMMATIC API

You can also import it as a library:

```
import { buildContext, runChecks } from 'dockerdoctor';

const context = await buildContext(process.cwd());
const report = await runChecks(context, {
  categories: ['dockerfile', 'secrets'],
  minSeverity: 'warning',
});

console.log(`Found ${report.summary.errors} errors`);
```

INSTALL

```
npm install -g dockerdoctor
# or just run it once:
npx dockerdoctor
```

Links:

- [GitHub](#)
- [npm](#)
- [r/docker_dev join the discussion](#)

It's MIT licensed, open source, and I'd genuinely love feedback. What checks would you want to see added? What Docker problems keep biting you?