Academia DevOps & Cloud · @jemjaf

DOCKER — CHECKLIST DE BUENAS PRÁCTICAS

> Para imágenes **seguras** y **mantenibles**. Encaja con la fase de contenedores del roadmap y cursos prácticos de la academia.

Orden **de menos a más complejo** de implementar (empieza arriba; lo de abajo suele llevar más prueba/error o cambios de proceso).

Nivel 1 — Cambios rápidos (minutos)

- ✅ `.dockerignore`: no mandar `node_modules`, `.git`, artefactos gordos ni secretos al contexto del build — menos contexto subido y builds más rápidos
- ✅ **Tag fijo por digest o versión** en `FROM` y al publicar: evitar `latest` sin control en producción
- ✅ **Logs a stdout/stderr** para que Docker / el orquestador los recoja (evitar logs "solo dentro del contenedor" sin salida estándar)

Nivel 2 — Comportamiento y seguridad base (algo de tiempo en Dockerfile / app)

- ✅ **No hardcodear** passwords ni API keys; env, Compose secrets u orquestador / vault según donde corras
- ✅ **Un proceso principal** por contenedor — salvo casos muy concretos; simplifica señales, reinicios y troubleshooting
- ✅ `HEALTHCHECK` (o probe equivalente fuera del Dockerfile) donde tenga sentido: que el sistema sepa si el servicio está "vivo" de verdad
- ✅ **Usuario no root** (`USER` en Dockerfile): ajustar ownership de rutas donde la app necesita escribir (aquí suele aparecer la fricción)

Nivel 3 — Imagen liviana, plataforma y hábitos de equipo (lo más delicado)

- ✅ **Límites de CPU/mem** en Compose / Kubernetes / runtime (nada "a infinito" en prod) — requiere acordar valores y revisar uso real
- ✅ **Base image mínima** (Alpine slim, distroless…) **cuando el runtime lo permita**: suele romper cosas sutiles (shell, libc, debugging) hasta que cierra
- ✅ **Multi-stage build**: compilar en una etapa; imagen final solo con binarios/recursos necesarios — reconstruye el Dockerfile entero hasta que cuadra
- ✅ **Política de actualización**: bases de imagen y dependencias (Renovate, rebuilds planificados, CI) — proceso continuo, no un solo comando

---

🌟 COMUNIDAD SKOOL — SIGUIENTE PASO

¿LISTO PARA PRACTICARLO DE VERDAD?

Esta guía es el mapa; la diferencia suele estar en **hacer labs sin quedarte atrapado ante un error** durante tres días. En mi comunidad de Skool aprendes con **cursos** dentro de la plataforma, **sesiones en vivo** y **soporte técnico** cuando tu entorno no levanta o tu despliegue falla.

👥 **Entra en la comunidad — skool.com/jemjaf** ↗️

Qué hay dentro (según la oferta vigente en Skool):

- 📚 **Cursos** grabados para ir a tu ritmo, más **lives** temáticos (cloud y DevOps)
- **Sandboxes temporales en AWS, Azure y GCP**: practica en nube **real** sin el miedo a una factura sorpresa
- 🛠️ **Soporte directo** para destrabar laboratorios, código y dudas puntuales
- 📈 **Career Center** para revisión de CV y LinkedIn con mirada técnica
- 📂 Materiales prácticos y guías que acompañan lo que enseño en clase

👉 **Únete en skool.com/jemjaf** ↗️ — Es formación hands-on y acompañamiento; sin prometer empleos automáticos. El objetivo es que **lleves práctica defendible**.