

Dockerfile Avanzado

Ejercicios Prácticos

Arturo Silvelo

Try New Roads

Ejercicios

Estos ejercicios prácticos te permitirán aplicar las técnicas avanzadas de Dockerfile aprendidas en los ejemplos. Trabajarás con aplicaciones reales:

- **Backend:** API REST desarrollada con **NestJS** (TypeScript)
- **Frontend:** Aplicación web desarrollada con **Angular** (TypeScript)

Cada ejercicio incluye un **Dockerfile base sin optimizar** que deberás mejorar aplicando las técnicas aprendidas.

Nota: Para ejecutar los ejercicios, asegúrate de estar en el directorio `ejercicios/` y seguir las instrucciones de cada sección.

Ejercicio 1: Backend NestJS

Objetivo

Optimizar progresivamente el Dockerfile del backend NestJS aplicando cada una de las técnicas vistas en el temario.

Aplicación

La aplicación backend está ubicada en `../../backend/` y contiene:

- API REST con NestJS
- Base de datos con TypeORM
- Autenticación JWT
- Documentación Swagger

1.1 Dockerfile Base

Analiza el `Dockerfile` sin optimizar ubicado en `backend/Dockerfile`.

Comando para construir:

```
docker build -f backend/Dockerfile -t backend-sin-optimizar ../../backend
```

1.2 Multi-stage Build

Crea `Dockerfile.multistage` que implemente multi-stage builds:

- **Etapas 1 (build):** Compila la aplicación TypeScript
- **Etapas 2 (test):** Ejecuta los tests sobre el código compilado
- **Etapas 3 (production):** Solo archivos necesarios para ejecutar

1.3 Optimización de Capas

Crea `Dockerfile.optimizado` mejorando el anterior:

- Agrupa comandos RUN relacionados
- Limpia archivos temporales en la misma capa
- Reordena instrucciones para mejor cache

1.4 Variables ARG y ENV

Crea `Dockerfile.variables` que gestione correctamente las variables:

- Usa **ARG** para las variables de entorno.

1.5 Gestión Segura de Secretos

Crea `Dockerfile.seguro` que elimine todos los secretos hardcodeados:

- Variables sensibles se pasan solo en runtime (`-e`)
- Opción para leer secretos desde archivos montados