

# Dockerfile Avanzado

---

**Arturo Silvelo**

Try New Roads

# Introducción a Docker Compose avanzado

---

Docker Compose es una herramienta esencial para definir y gestionar aplicaciones multicontenedor en Docker. Permite describir la arquitectura de una aplicación, sus servicios, redes y volúmenes en un solo archivo YAML, facilitando la orquestación y el despliegue en diferentes entornos.

## Aliases en YAML

---

En archivos YAML, como los usados por Docker Compose, los anchors (`&`) y alias (`*`) permiten reutilizar bloques de configuración, evitando duplicidad y facilitando el mantenimiento.

- **Anchor (`&`)**: Define un bloque reutilizable.
- **Alias (`*`)**: Inserta el bloque definido por el anchor.
- **Merge (`<<`)**: Permite combinar configuraciones.

## Ventajas

---

- Menos repetición de código.
- Cambios centralizados y más fáciles de mantener.
- Configuraciones más limpias y legibles.

# Healthcheck

---

El parámetro `healthcheck` en Docker Compose permite definir una comprobación periódica para saber si un servicio está funcionando correctamente. Esto ayuda a detectar fallos y a gestionar dependencias entre servicios.

El parámetro `healthcheck` puede definirse tanto en el `Dockerfile` como en cada servicio dentro de Docker Compose.

En ambos casos, se dispone de varias opciones de configuración:

- **test**: Comando que se ejecuta para comprobar la salud.
- **interval**: Frecuencia de la comprobación.
- **timeout**: Tiempo máximo de espera para la comprobación.
- **retries**: Número de intentos antes de marcar el servicio como unhealthy.
- **start\_period**: Tiempo de gracia antes de empezar a comprobar.

## `depends_on`

---

Es posible emplear `depends_on` junto con la condición `service_healthy` para garantizar que un servicio espere a que otro esté en estado saludable antes de iniciar su ejecución.

El servicio dependiente será creado y su contenedor iniciado, pero no comenzará su proceso principal hasta que el servicio del que depende alcance el estado saludable (`healthy`).

## Gestión de entornos

---

En el desarrollo de aplicaciones, un entorno define el contexto en el que se ejecuta la aplicación: desarrollo, pruebas, integración o producción. Cada entorno puede requerir configuraciones, variables y servicios diferentes.



Docker y Docker Compose permiten gestionar estos entornos de varias formas:

- **Variables de entorno:** Permiten parametrizar el comportamiento de los servicios según el entorno. Se pueden definir directamente en el archivo Compose, en archivos `.env` o pasarlas desde el sistema.
- **Archivos Compose específicos:** Puedes crear archivos `docker-compose` adicionales (por ejemplo, `docker-compose.dev.yaml`, `docker-compose.prod.yaml`) para sobrescribir o extender la configuración base según el entorno.

- **Múltiples Dockerfile:** Es posible mantener diferentes Dockerfile (por ejemplo, Dockerfile.dev, Dockerfile.prod) adaptados a las necesidades de cada entorno, seleccionando el adecuado en la sección build del Compose.

Estas estrategias permiten adaptar fácilmente la infraestructura y el despliegue de la aplicación a cada fase del ciclo de vida.