



Nube de Alibaba [Seguir](#)

Una compañía global de computación en la nube y AI, una subsidiaria de Alibaba Group. Sitio web: <https://www.alibabacloud.com/>  
12 de diciembre · Lectura de 7 minutos

## Cambiando la forma de entrega continua con Docker (Parte 2)



### Introducción

Docker es un servicio que ha reformado la forma de realizar la entrega continua. Docker le permite empaquetar una aplicación con todas sus dependencias en una unidad estandarizada para el desarrollo de software y garantiza que el software siempre funcionará igual, independientemente de su entorno. La primera parte de esta serie proporcionó un resumen de la descripción general, los desafíos y el entorno y los requisitos de construcción para usar Docker. Esta publicación explorará los métodos específicos para usar Docker junto con los procesos de entrega.

### Cómo usar Docker

### Paso 1: instalación del entorno de tiempo de ejecución de Docker

El entorno de tiempo de ejecución de Docker tiene solo una limitación, es decir, el kernel de operación de Linux debe ser más alto que la versión 3.10. Docker proporciona un método conveniente para instalar y configurar un entorno Docker en su máquina utilizando Docker Toolbox . Este instalador le permite instalar fácilmente Docker independientemente del sistema operativo subyacente. También es compatible con una variedad de servidores en la nube, incluidos los servidores del servicio de computación elástica (ECS) de Alibaba Cloud .

## Paso 2: Usar Dockerfile para describir el entorno de la aplicación

- Método 1:

El personal de O & M debe enumerar todos los elementos originalmente requeridos para construir un entorno y describirlos con el archivo Docker. Dockerfile no es un lenguaje de desarrollo, sino solo un lenguaje de descripción definido. Para crear un lenguaje de descripción, debe incluir la instrucción **FROM** y un entorno en **FROM** . Cuando esto se define, el usuario puede realizar tareas con los roles y responsabilidades asignados, incluyendo correr para ejecutar comandos, evento como una variable de entorno, así como **agregar** , **exponer** y comandos de **cmd** .

- Método 2:

Tras el establecimiento exitoso de un entorno Java estándar, uno puede usarlo directamente aplicando **jar** a él.

Si los servicios se desmontan en micro-servicios (cada servicio no es un paquete de **guerra** sino un **contenedor** ), entonces todos los micro-servicios deberían compartir una imagen común de Docker. Muchos idiomas tienen la función de API, y en la mayoría de los casos no requieren dependencias, pero requieren el entorno de ejecución del idioma.

Las imágenes de Docker para entornos variados están disponibles en muchos sitios web oficiales. Puede usar imágenes del sitio web oficial **FROM** directamente aplicando **jar** a ellas. Además, algunos programas de código abierto permiten al usuario ejecutarlos directamente después de su descarga, como Gitlab. Con la tecnología Docker, también es mucho más fácil construir varios sistemas de herramientas de código abierto.

## Paso 3: Uso de Docker para compilar entornos de compilación / pruebas de unidades

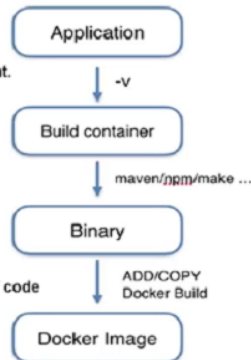
### Step 3: Describe your build / UT environment with Docker

Benefits of using Docker as build / UT environment:

- User-defined, extensible, duplicable build environment.

Point:

- Build app and build docker image are carried out separately
  - The image layering concept may result in the leakage of source code
  - Image minimization principle
- Docker file must not be put under the root directory of code
  - Avoid a large number of files sent to docker daemon
- Run the test script directly through runtime container



- **Compilación separada de aplicaciones y entornos Docker**

Al compilar aplicaciones y entornos Docker juntos, es necesario copiar el código de la aplicación en el entorno Docker. Sin embargo, durante el proceso de construcción de un entorno Docker, si el código de la aplicación se genera dentro del entorno Docker, se eliminarán. Dada la estructura jerárquica de Docker (con mayúsculas armonizadas y minúsculas a lo largo de un contexto), cualquier `.java` copiado al archivo Docker para *compilar* un `jar` se eliminará después de que se *crea* el `jar`. Dado que el acceso al código fuente es simplemente a través de `.java`. Además de esto, el código no es deseable en un entorno de producción cuando el contenedor Docker se ejecuta allí.

- **Separación del entorno de compilación y prueba**

Basado en el principio mínimo de imagen de Docker (solo instala lo que es esencial), un entorno en ejecución Java (mayúsculas y minúsculas armonizadas) + NodeJS no requiere ping después de que un entorno de producción comience a ejecutarse, y solo es necesario para la depuración durante O & M. La compilación puede depender mucho más del entorno de compilación, pero la ejecución de un entorno de producción no requiere dependencias. Por lo tanto, recomendamos separar los entornos y crear un entorno de compilación también mediante el uso de Docker.

Un entorno de compilación es solo para compilar, mientras que un entorno de prueba solo se ejecuta. El proceso de usar un entorno para

procedimientos de construcción sigue este simple principio:

*La carga de código puede continuar utilizando el método -v en un contenedor llamado "compilación", donde puede usar diferentes idiomas para compilar archivos como paquetes de dependencia binarios; luego agregue estos archivos al archivo Docker para la compilación Docker. A través de los dos pasos de construcción, construirá con éxito las imágenes de Docker que se pueden ejecutar.*

Además del código fuente, también se ejecuta el binario creado a través del primer paso de creación y agregado al archivo Docker.

Después de crear una imagen, puede activar diferentes scripts, como activar un script de producción para ejecutar la producción o activar un script de prueba para ejecutar una prueba. Puede usar Docker para ejecutar una prueba integrada también. Para una aplicación Dockerized, una vez que se ejecuta una Docker para ejecutar un comando de prueba, se ejecutará un proceso de prueba, independientemente de si se trata de una base de datos o no. Todo lo que se pueda escribir para las pruebas se puede escribir con Docker.

## Paso 4: Uso de Docker para describir dependencias en el entorno

Una aplicación no puede hacer solo con imágenes sino que también depende de una base de datos, si no de otra aplicación. Por ejemplo, una base de datos es necesaria cuando se realizan pruebas de datos.

Docker también ofrece una herramienta llamada Docker Compose, que puede administrar múltiples imágenes de Docker. Consulte los siguientes pasos para usar Docker Compose:

1. **Escriba un archivo Docker para describir un entorno de aplicación**

Describe qué es un directorio de proyecto en el archivo Docker, copie un archivo de requisitos en y bajo el directorio de códigos, instale un programa ruby y ejecute el programa.

2. **Utilice .yaml para describir la dependencia**

En la mayoría de los casos, usar un solo contenedor no resuelve la mayoría de los problemas. Puede usar .yaml para describir una o más aplicaciones web, bases de datos y otros.

Cuando una aplicación necesita depender de una base de datos, puede escribir un enlace y ponerle el nombre "link DB". Al utilizar el comando

Docker Compose, podrá ejecutar un grupo de imágenes de Docker, y podrá describir las dependencias que contiene con `compose .yaml`.

1. **Use Docker para describir la integración completa o el entorno en ejecución.**

Después de Docker, la aplicación puede necesitar un equilibrador de carga, base de datos, almacenamiento de archivos, caché u otros elementos.

- **A**

Docker de **base de datos** no le importa la ubicación de creación de una base de datos o los medios para su creación. Durante un proceso de control de desarrollo, si necesita iniciar un DB, tendrá que extraer el contenedor de DB de la fuente oficial correspondiente. Por ejemplo, cuando necesite utilizar un MySQL DB5.7, inicie el contenedor de MySQL DB para ver la configuración correspondiente, o cuando sea necesario, para modificar algunas descripciones de idioma. Puede ejecutarlo en Docker Compose. Escribir dependencias en la primera línea para ejecutar MySQL 5.7; escribir variables ambientales en la segunda línea para reemplazar los archivos de configuración por los suyos propios; configure la contraseña de inicio y luego inicie el puerto en 3306 o de lo contrario.

Para probar una aplicación web, no es necesario que escriba una ID de MySQL, y solo MySQL puede hacer lo que MySQL describe en "redactar". Luego obtenga acceso a la base de datos ingresando la contraseña. Para Redis, solo necesita nombrarlo y proporcionarle el puerto, y solo necesita el servicio de caché.

- **El equilibrio de**

carga de **Load Balancer** también es simple, y tanto nginx como Haproxy ofrecen herramientas oficiales. La mayor preocupación para los desarrolladores es utilizar Docker para describir sus aplicaciones. Después de completar una descripción, ejecute la aplicación completando la "redacción" y especificando la versión y configuración de nginx.

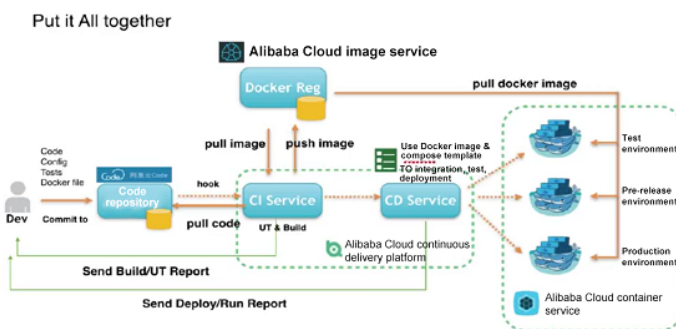
- **Almacenamiento**

Monte en la máquina host usando `-v`, y luego almacene la base de datos y los datos inicializados en un directorio local. Esto le permite hacer clic en la aplicación en lugar de inicializarla cada vez que se ejecuta.

Durante la ejecución de un entorno de producción, el almacenamiento de datos se realiza en un servidor en la nube, independientemente de la implementación. No importa si ha utilizado NAS o Alibaba Cloud Object Storage Services (OSS), puede armar un directorio en un directorio local, en lugar de cargar los datos mediante SDK o API a un dispositivo de almacenamiento en la nube.

## Proceso de entrega dockerized

### Complete continuous delivery process of containers



Además de los scripts de código, configuración y prueba, los desarrolladores también deben escribir Dockerfile. Al presionar el código en un repositorio de código, el servicio de CI le dirá una nueva confirmación a través del enlace de depósito de código. A continuación, extraerá y copiará el Código para compilación y UT. Durante el proceso de compilación, extraerá la imagen de la que depende del Registro Docker; una vez que el proceso de compilación está completo, enviará la imagen a la unidad de registro de Docker. Entonces, CI tendrá un gancho para informar al CD, y el Servicio de implementación desplegará la imagen en un entorno de prueba, prueba o producción basado en Docker, la descripción de la imagen y la descripción de la composición.

Todo el proceso de implementación consiste en extraer una imagen de compilación del Registro y ejecutarla. Dado que el contenido más complicado se encuentra en la imagen y los archivos de "redacción", todo el proceso se simplifica.

## Conclusión

### 1. ¿Qué trae la entrega de Docker?

Hace que el desarrollo sea más claro y más flexible para los

desarrolladores. Anteriormente, la preocupación estaba en la dependencia de Java, y era bastante difícil construir un entorno Java localmente. Sin embargo, ahora que la preocupación se ha desplazado a la dependencia de todo el entorno, hace que el entorno Java sea mucho más fácil de construir. Con este enfoque, el personal de O & M ya no necesita cambiar su horario debido a dependencias de software, y no tiene que cambiar constantemente la configuración. ¿A qué se puede aplicar Docker? Cada proyecto y entorno para todos. También es la implementación más efectiva de DevOps. Con Docker, puede aplicar la programación a las operaciones y administrar las operaciones ambientales utilizando prácticas de desarrollo.

## **2. ¿Qué escenarios deberían tener Dockerized?**

La definición completa relacionada con esto aún está por surgir. Sin embargo, entiendo que el mayor beneficio de la conversión en un contenedor es la realización de ciclos de vida más cortos y la apatridia. Por ejemplo, para aplicaciones web como REST API o CI / CD, puede extraer todo el contenedor Docker para ejecutar durante una prueba, y luego descartarlo una vez que se complete la prueba para evitar la ocupación de recursos. En un proceso de realización de tareas, puede ahorrar en recursos utilizando Docker.

Las bases de datos, especialmente las de almacenamiento, son las más difíciles de copiar y, por lo tanto, pueden no beneficiarse directamente de los contenedores. Si un servidor soporta una carga mucho mayor, sirve como almacenamiento de archivos. Sin embargo, esto no significa que no sea factible, pero puede ser de poca importancia.







