



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 · 6 minutos de lectura

Cambiando la forma de entrega continua con Docker (Parte 1)



Introducción

Esta publicación es la primera parte de la serie "**Cambio de la forma de entrega continua con Docker**" y analiza los antecedentes, desafíos y procesos relacionados con Docker. Docker es un servicio para entrega continua reformada. En la segunda parte de la serie, exploraremos el método de utilizar Docker junto con sus procesos de entrega.

Fondo

La industria de Internet ha cambiado constantemente las demandas y los productos del mercado, lo que ha obligado a las organizaciones a adaptarse haciendo entregas constantes y actualizaciones en su entorno de producción. Este nuevo enfoque del desarrollo se conoce como Integración Continua (CI) y Entrega Continua (CD), que combina y convierte los procesos de desarrollo, producción y entrega en un proceso cíclico.

Sin embargo, este enfoque introduce varios problemas a largo plazo. Uno de estos problemas es la dificultad de transferir ambientes de

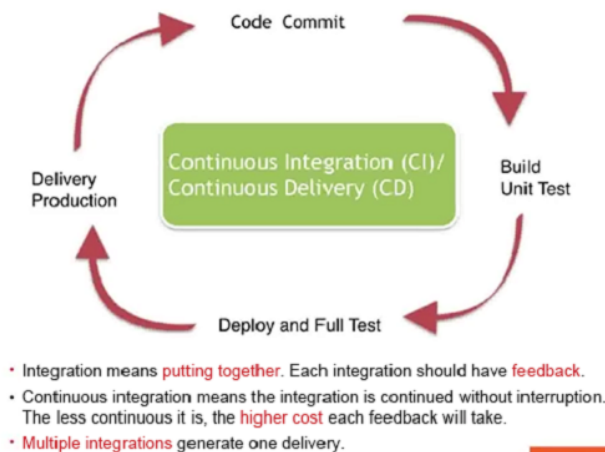
producción oscuros a sucesores que tienen una experiencia limitada con los entornos.

Además, la eliminación de errores que ocurre en el entorno de producción antes del lanzamiento final de la producción es un obstáculo. No solo es difícil de mantener, sino que también requiere actualizaciones constantes porque las demandas y los productos cambian constantemente.

Procesos tradicionales de CD: visión general y desafíos

Las soluciones de desarrollo tradicionales incluyen procesos, integración continua (CI) y entrega continua (CD). Estas prácticas combinan y convierten los procesos de desarrollo, producción y entrega en un proceso cíclico.

Solution



- **Integración** (combinando dos cosas juntas): Tras la presentación de los documentos, el código va con el código; al compilar, el código va con la lógica; después de las pruebas, el código va con características; desde la generación hasta la implementación y luego de la generación para su lanzamiento, el código va con los sistemas y los sistemas van con los sistemas. Cada vez que el sistema combina dos cosas, ocurre la integración.
- **Pruebas continuas** : al igual que un examen físico (para su sistema), ayuda con el descubrimiento rápido y la eliminación de fallas en su sistema antes del lanzamiento final, evitando una combinación de características no probadas, características faltantes o segmentos de código defectuosos. En general, este proceso debe ser continuo.

- **Bucles de retroalimentación** : en el diagrama de flujo de los tubos de integración continua, cada paso consiste en un ciclo de retroalimentación. Esto permite una retroalimentación rápida, simplificando la identificación y rectificación de problemas.

Por ejemplo, después de enviar el código, un desarrollador primero debe asegurarse de que el código no entre en conflicto con otros códigos. El desarrollador también prueba si el código se compila exitosamente para pasar la prueba unitaria. Después de cumplir con todos estos requisitos, el desarrollador puede pasar al siguiente elemento. Sin embargo, si el desarrollador envió un código no conflictivo pero recibe comentarios de que el código inhabilitó la compilación exitosa de todo el sistema, el desarrollador debe corregir el código.

Los comentarios deben rastrearse rápidamente hasta el punto de desarrollo para que los desarrolladores puedan saber qué hacer a continuación. Además, las pruebas unitarias deben mantenerse separadas de la función del sistema o las pruebas de integración. Esto se debe a que la velocidad de estas pruebas varía, y las pruebas generan diferentes tipos de comentarios.

Creación de un proceso de CD: requisitos del entorno

En general, una empresa o un proyecto requiere un entorno múltiple. Normalmente, el entorno de producción de una empresa se coloca en una nube pública, mientras que el proceso de desarrollo se realiza en un entorno fuera de línea. El entorno de nube pública puede ser inconsistente con el entorno de desarrollo fuera de línea, lo que genera problemas durante el lanzamiento del producto final.

Procesos de CD: problemas encontrados

Pueden surgir problemas incluso si se genera sistemáticamente un entorno de sistema de integración continua completo. Los desarrolladores pueden incluso confiar en diferentes entornos o paquetes de idioma, lo que genera conflictos en entornos de compilación y dificultades en el mantenimiento.

Problems in the CD process

- Difficult to maintain the build environment
- Difficult to maintain the dependent environment
- Difficult to switch the environment for debugging
- Difficult to maintain the version of runtime package
- Even more difficult to unify environment criteria and trace back environment

Origen de los problemas

La mayoría de los problemas surgen porque los desarrolladores solo entregan código y dependencias relacionadas, mientras que las operaciones, en realidad, también requieren un entorno operativo, descripción ambiental, dependencias, bases de datos y caché.

The root cause of the problem is that the developer only delivers code

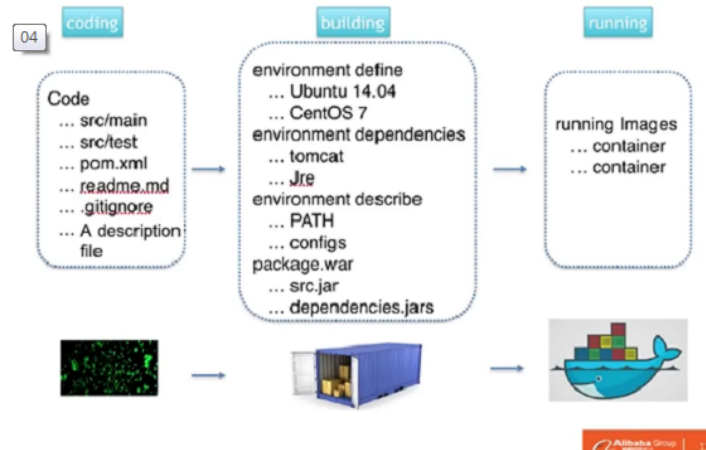


Picture source: <https://www.enterpriseitmagazine.com/wp-content/uploads/2011/09/devops.jpg>

- The developer only delivers code and code dependencies
- To keep a site running, we need not only the **operating environment** in addition to code, but also the **dependencies** between operating environments

Docker: Transformando la forma de entrega de software

If we can deliver the runtime stack of the whole software easily



En Docker, toda la información requerida para la generación del entorno está incluida en el código entregado. El código contiene un archivo de descripción que describe el entorno, así como sus dependencias, cachés, configuraciones, variables, contenedores y paquetes jar.

Este enfoque es análogo a empaquetar un código y todos los demás componentes necesarios en un contenedor, y entregar el contenedor al equipo de operaciones. La característica central de este contenedor es su portabilidad. Dado que el contenedor incluye todas las dependencias ambientales, podemos obtener el mismo resultado independientemente del entorno operativo.

Competencia de Docker

Antes de que apareciera Docker, había muchas limitaciones de especificación para crear contenedores. Docker no es solo un software sino también un nuevo enfoque para implementar contenedores.

- **Capacidad de descripción del entorno** : los archivos de Docker pueden describir todo el entorno requerido por el software.
- **Sistema de archivos jerárquico** : las imágenes de Docker proporcionan una solución para la gestión de paquetes, en la que describimos cada operación como una capa de gestión de versiones.
- **Separación del sistema operativo** : Docker protege las diferencias del sistema operativo al ejecutarse.

Docker es una tecnología de contenedores

En la tecnología de virtualización, el hardware y el software se virtualizan en máquinas virtuales. Cada máquina virtual es un sistema operativo completo que está bien aislado, pero puede requerir varios minutos para comenzar.

La principal diferencia de la tecnología de contenedor de la tecnología de virtualización tradicional radica en el hecho de que todos los contenedores carecen de una capa de sistema operativo completa y utilizan el núcleo del sistema operativo de la máquina principal como propio.

La principal ventaja de la tecnología de contenedores es que los contenedores se pueden iniciar en segundos ya que no involucran ninguna máquina virtual tradicional. Además, dado que los contenedores tienen una sobrecarga menor en comparación con las máquinas virtuales, los usuarios pueden implementar más contenedores en un servidor.

Tres pasos para la entrega de software con Docker

1. **Build** : describe la base del sistema operativo, el entorno, el puerto para iniciar y las secuencias de comandos para ejecutar. El sistema guarda el archivo de descripción como una imagen Docker ubicada dentro del almacenamiento local.
2. **Enviar** : empuja la imagen al Registro de Docker en el otro extremo.
3. **Ejecutar** : extrae la imagen del registro público al ejecutarse. El contenedor es una descripción ambiental y, al mismo tiempo, una totalidad, que ofrecerá el mismo resultado cuando se ejecute en cualquier entorno.

Estudio de caso: BBC News

BBC News es una compañía global de noticias del sitio web con más de 500 desarrolladores distribuidos por todo el mundo. Tiene más de diez entornos de CI, ya que utiliza diferentes idiomas en diferentes áreas del mundo. BBC News tuvo que descubrir cómo unificar los procesos de codificación y administrar los entornos de CI de manera uniforme. Los trabajos existentes tomaron hasta 60 minutos para programarse y ejecutarse, y se ejecutaron de forma secuencial. Con Docker, los trabajos ahora se ejecutan en paralelo, acelerando significativamente el proceso. Además, al usar contenedores, los desarrolladores no tienen

que preocuparse por los entornos de CI. [Visite Docker para obtener más información sobre la historia de éxito de BBC](#).

Conclusión

Esta publicación presentó Docker y su papel en la integración continua y la entrega continua. La entrega continua con Docker se centra principalmente en la reducción de los riesgos de las aplicaciones a la vez que ofrece un valor más rápido a través de la producción de software confiable en iteraciones más cortas.

En la siguiente parte de esta publicación, examinaremos cómo usar Docker y describir su construcción y entorno UT.

