Comience, Parte 1: Orientación y configuración

Tiempo estimado de lectura: 4 minutos

¡Bienvenido! Estamos emocionados de que quieras aprender Docker. El *tutorial Docker Get Started le* enseña a:

1. Configure su entorno Docker (en esta página)
2. Crea una imagen y ejecútala como un contenedor (https://docs.docker.com/get-started/part2/)
3. Escala tu aplicación para ejecutar varios contenedores (https://docs.docker.com/get-started/part3/)
4. Distribuya su aplicación a través de un clúster (https://docs.docker.com/get-started/part4/)
5. Apila los servicios agregando una base de datos back-end (https://docs.docker.com/get-started/part5/)
6. Implementa tu aplicación en producción (https://docs.docker.com/get-started/part6/)

Conceptos de Docker

Docker es una plataforma para desarrolladores y administradores de sistemas para **desarrollar, implementar y ejecutar** aplicaciones con contenedores. El uso de contenedores Linux para implementar aplicaciones se denomina *contenedorización* . Los contenedores no son nuevos, pero su uso para implementar aplicaciones fácilmente sí lo es.

La contenedorización es cada vez más popular porque los contenedores son:

- Flexible: incluso las aplicaciones más complejas pueden contenerse.
- Ligero: los contenedores aprovechan y comparten el kernel de host.
- Intercambiable: puede implementar actualizaciones y actualizaciones sobre la marcha.
- Portátil: puedes construir localmente, implementar en la nube y ejecutar en cualquier lugar.
- Escalable: puede aumentar y distribuir automáticamente réplicas de contenedores.
- Apilable: puede apilar servicios verticalmente y sobre la marcha.

Imágenes y contenedores

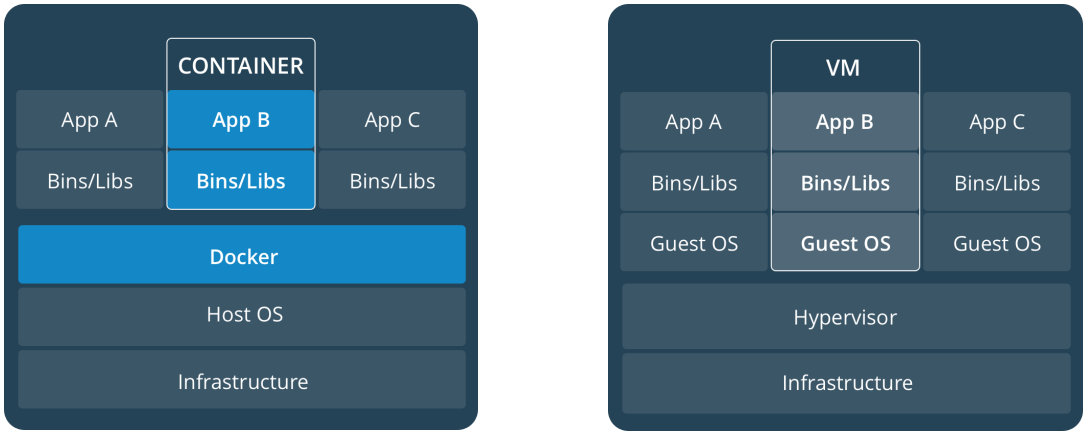
Un contenedor se inicia ejecutando una imagen. Una **imagen** es un paquete ejecutable que incluye todo lo necesario para ejecutar una aplicación: el código, un tiempo de ejecución, bibliotecas, variables de entorno y archivos de configuración.

Un **contenedor** es una instancia en tiempo de ejecución de una imagen: qué imagen se convierte en memoria cuando se ejecuta (es decir, una imagen con estado o un proceso de usuario). Puede ver una lista de sus contenedores en ejecución con el comando `docker ps` , tal como lo haría en Linux.

Contenedores y máquinas virtuales

Un **contenedor se** ejecuta *nativamente* en Linux y comparte el kernel de la máquina host con otros contenedores. Se ejecuta un proceso discreto, no teniendo más memoria que cualquier otro ejecutable, por lo que es ligero.

Por el contrario, una **máquina virtual** (VM) ejecuta un sistema operativo "invitado" completo con acceso *virtual* a recursos de host a través de un hipervisor. En general, las VM proporcionan un entorno con más recursos de los que la mayoría de las aplicaciones necesitan.



Prepare su entorno Docker

Instale una versión mantenida (https://docs.docker.com/engine/installation/#updates-and-patches) de Docker Community Edition (CE) o Enterprise Edition (EE) en una plataforma compatible (https://docs.docker.com/engine/installation/#supported-platforms) .

> **Para la integración completa de Kubernetes**
>
> - Kubernetes on Docker para Mac (https://docs.docker.com/docker-for-mac/kubernetes/) está disponible en 17.12.0-ce Edge (https://docs.docker.com/docker-for-mac/release-notes/#docker-community-edition-17120-ce-mac45-2018-01-05-edge) o superior.
> - Kubernetes en Docker para Windows (https://docs.docker.com/docker-for-windows/kubernetes/) está disponible en 18.02.0-ce Edge (https://docs.docker.com/docker-for-windows/release-notes/#docker-community-edition-18020-ce-rc1-win50-2018-01-26-edge) o superior.

Instalar Docker (https://docs.docker.com/engine/installation/)

Versión Docker de prueba

Asegúrese de tener una versión compatible de Docker:

```
$ docker --version
Docker version 17.12.0-ce, build c97c6d6
```

Ejecute `docker version` (sin `--`) o `docker info` para ver aún más detalles sobre la instalación de su docker:

```
$ docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 17.12.0-ce
Storage Driver: overlay2
...
```

> **Nota** : Para evitar errores de permiso (y el uso de `sudo`), agregue su usuario al `docker` grupo. Leer más (https://docs.docker.com/engine/installation/linux/linux-postinstall/) .

Instalación de Test Docker

Pruebe que su instalación funcione ejecutando la imagen Docker simple, hello-world (https://hub.docker.com/_/hello-world/) :

```
$ docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:ca0eeb6fb05351dfc8759c20733c91def84cb8007aa89a5bf606bc8b315b9fc7
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
...
```

Enumere la `hello-world` imagen que se descargó en su máquina:

```
$ docker image ls
```

Haga una lista del `hello-world` contenedor (generado por la imagen), que se cierra después de mostrar su mensaje. Si todavía estuviera ejecutándose, *no* necesitarías la `--all` opción:

```
$ docker container ls --all
CONTAINER ID     IMAGE           COMMAND      CREATED            STATUS
54f4984ed6a8     hello-world     "/hello"     20 seconds ago     Exited (0) 19 seconds ago
```

Resumen y hoja de trucos

```
## List Docker CLI commands
docker
docker container --help

## Display Docker version and info
docker --version
docker version
docker info

## Excecute Docker image
docker run hello-world

## List Docker images
docker image ls

## List Docker containers (running, all, all in quiet mode)
docker container ls
docker container ls -all
docker container ls -a -q
```

Conclusión de la primera parte

La contención hace que CI / CD sea (https://www.docker.com/use-cases/cicd) transparente. Por ejemplo:

- las aplicaciones no tienen dependencias del sistema
- las actualizaciones se pueden enviar a cualquier parte de una aplicación distribuida

- la densidad de recursos puede ser optimizada.

Con Docker, escalar su aplicación es una cuestión de girar nuevos ejecutables, no ejecutar hosts de VM pesados.

En la parte 2 »
[https://docs.docker.com/get-started/part2/]

get started (https://docs.docker.com/glossary/?term=get%20started) , setup (https://docs.docker.com/glossary/?term=setup) , orientation (https://docs.docker.com/glossary/?term=orientation) , quickstart (https://docs.docker.com/glossary/?term=quickstart) , intro (https://docs.docker.com/glossary/?term=intro) , concepts (https://docs.docker.com/glossary/?term=concepts) , containers (https://docs.docker.com/glossary/?term=containers)