

Isma's Blog

Blog sobre tecnologías de desarrollo

Creando una imagen de Ubuntu con Docker

Introducción

En este artículo mostraré cómo crear una imagen Docker que contenga Ubuntu 16.04 a partir de una imagen ya existente. Actualizaré la imagen descargada y crearé una nueva imagen con Ubuntu actualizado. Es una tontería, pero me sirve para probar ciertos comandos docker, y ver cómo modificar una imagen y subirla a mi cuenta Docker Hub.

Buscar la imagen de Ubuntu

Primero buscamos las imágenes que contengan la palabra ubuntu con el comando **search**

```
$ docker search ubuntu
```

Después de ejecutar el comando, se debería mostrar un listado de imágenes docker parecido al que se muestra en la siguiente imagen:

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating s...	3843	[OK]	
ubuntu-upstart	Upstart is an event-based replacement for ...	61	[OK]	
centoschep/ubuntu-sshd	Dockerized SSH service, built on top of of...	25		[OK]
torusware/speeduz-ubuntu	Always updated official Ubuntu docker inag...	25		[OK]
ubuntu-debootstrap	debootstrap --variant=minbase --components...	23	[OK]	
nickistre/ubuntu-lamp	LAMP server on Ubuntu	6		[OK]
nickistre/ubuntu-lamp-wordpress	LAMP on Ubuntu with wp-cli installed	5		[OK]
nimmis/ubuntu	This is a docker images different LTS vers...	4		[OK]
imagebec/ubuntu	Simple always updated Ubuntu docker images...	4		[OK]
maxexloo/ubuntu	Docker Base image built on Ubuntu with Sup...	2		[OK]
adniringwara/ubuntu	Base ubuntu images based on the official u...	1		[OK]
jordii/ubuntu	Ubuntu Base Image	1		[OK]
larksheer/ubuntu	Base Ubuntu Image -- Updated hourly	1		[OK]
ssycat/ubuntu	Ubuntu LTS	0		[OK]
konstruktoid/ubuntu	Ubuntu base image	0		[OK]
dehhyppie/ubuntu	Docker images for ubuntu	0		[OK]
lynxtp/ubuntu	https://github.com/lynxtp/docker-ubuntu	0		[OK]
life360/ubuntu	Ubuntu is a Debian-based Linux operating s...	0		[OK]
callias/ubuntu	Ubuntu with the needful	0		[OK]
viderplan/ubuntu	Our basic Ubuntu images.	0		[OK]
datenbetrieb/ubuntu	custom flavor of the official ubuntu base ...	0		[OK]
teamrock/ubuntu	TeamRock's Ubuntu image configured with AU...	0		[OK]
ustclug/ubuntu	ubuntu image for docker with USIG mirror	0		[OK]
uvatbc/ubuntu	Ubuntu images with unprivileged user	0		[OK]
lorapro/ubuntu	ubuntu image	0		[OK]

Descargar la imagen de Ubuntu

El siguiente paso sería descargar la imagen Docker que hayamos elegido. En este ejemplo elijo la imagen con el nombre **ubuntu** que es la que más estrellas tiene, y dentro de esa imagen, elijo la última versión indicando **latest**. Para ello utilizamos el comando **pull**:

```
$ docker pull ubuntu:latest
```

Después de ejecutar el comando, nos debería aparecer en la consola un resultado parecido al que se muestra en la siguiente imagen.

```
$ docker pull ubuntu:latest
latest: Pulling from library/ubuntu
6d28225f8d96: Pull complete
166102ec41af: Pull complete
d09bfb2bd6a: Pull complete
c80dad39a6c0: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:5718d664299eb1db14d87db7bfa6945b28879a67b74f36da3e34f5914866b71c
Status: Downloaded newer image for ubuntu:latest
```

Si ejecutamos el comando **images**, la nueva imagen debería aparecer en el listado

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	c5f1cf30c96b	5 days ago	120.8 MB
hello-world	latest	94df4f0ce8a4	12 days ago	967 B
busybox	latest	47bcc53f74dc	7 weeks ago	1.113 MB

Creación del contenedor

Una vez tenemos la imagen en nuestro repositorio local de imágenes Docker, creamos el contenedor. Esto se lleva a cabo ejecutando el comando **run**:

```
$ docker run -i -t <IMAGE-ID>
```

donde <IMAGE-ID> es el campo IMAGE ID del listado de imágenes Docker de nuestra imagen bajada recientemente.

En este caso, al crearse el contenedor, se nos crea una sesión bash a partir de la cual ya podríamos trabajar como si estuviésemos en la máquina Ubuntu.

Si queremos parar el contenedor, escribimos **exit** y el contenedor se para y vuelve al cliente Docker.

Ejecutando el comando docker **ps**, vemos los contenedores de nuestro sistema Docker, y nos fijamos en la columna CONTAINER ID de nuestra imagen Docker:

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c22f2507909	c5f1cf30c96b	"/bin/bash"	4 minutes ago	Exited (0) 2 minutes ago		gigantic_pansaujan
c742f90c320	c5f1cf30c96b	"/bin/bash"	6 minutes ago	Exited (0) 4 minutes ago		dreamy_ginoussi
86ea4f0c35d6	c5f1cf30c96b	"/bin/bash"	12 minutes ago	Exited (0) 8 minutes ago		furious_kalan
18c7e44c42e	c5f1cf30c96b	"/bin/bash"	12 minutes ago	Exited (0) 12 minutes ago		big_davein
47008daaf7e	busybox	"sh"	23 hours ago	Exited (0) 23 hours ago		jelly_archimedes
819e82e08054	busybox	"sh"	24 hours ago	Exited (0) 24 hours ago		silly_boyd
ee40015513f6	busybox	"sh"	24 hours ago	Exited (0) 24 hours ago		evil_puninsky
7d4d0b5b15b0	busybox	"sh"	25 hours ago	Exited (0) 24 hours ago		happy_dubinsky
bb57fb2263d0	hello-world	"/hello"	25 hours ago	Exited (0) 25 hours ago		agitated_swirlins
244af08ec4ef	hello-world	"/hello"	26 hours ago	Exited (0) 26 hours ago		thirsty_dewinci
ad06ea3473d	hello-world	"/hello"	2 days ago	Exited (0) 2 days ago		glenny_khorana

Con el CONTAINER_ID podemos volver a arrancar nuestro contenedor con el comando **start**

```
$ docker start ca2f295d9b89
```

Si ejecutamos de nuevo el comando **ps**, comprobaremos que el contenedor está levantado.

Ahora, podemos "incluínos" en el contenedor con el comando **attach**

```
$ docker attach ca2f295d9b89
```

y estaremos dentro de la instancia del contenedor.

En este punto, por ejemplo, voy a actualizar Ubuntu ejecutando **apt-get update**.

Grabando la nueva imagen

Una vez hayamos instalado en nuestro contenedor todas las herramientas que creamos necesarias para tener listo el contenedor, podemos guardar el contenedor como una nueva imagen para en un futuro utilizarla.

Para guardar la instancia del contenedor utilizamos el comando **commit <CONTAINER ID> [REPOSITORY:TAG]**

```
$ docker commit ca2f295d9b89 ubuntu:16.04
```

Si listamos las imágenes Docker, veríamos la imagen recién creada con el **TAG 16.04**.

Subiendo la imagen a Docker Hub

Para poder subir imágenes creadas por nosotros debemos tener una cuenta en Docker Hub.

Con una cuenta creada, antes de enviar nuestra imagen debemos logarnos en el cliente Docker con el comando **login**

```
$ docker login
```

Este comando nos pedirá el usuario y contraseña de nuestra cuenta Docker Hub y ya podríamos poder enviar nuestras imágenes con el comando **push**

```
$ docker push ismaelcaban/ubuntu
```

Para subir una imagen a nuestra cuenta Docker, la imagen debemos crearla con el formato **identificador_cuenta_docker_hub/image_name**.

Conclusión

Hemos visto, de una forma general, el proceso de bajarnos una imagen de Docker Hub, modificarla a nuestro gusto y subirla a nuestra cuenta Docker para tenerla disponible en cualquier momento.

Anuncios

Esta entrada se publicó en Docker, Integración Continua y está etiquetada con Docker, integración continua, ubuntu en mayo 12, 2016 [<https://pajarakillo.wordpress.com/2016/05/12/creando-una-imagen-de-ubuntu-con-docker/>] .

