

Desplegament d'aplicacions amb Docker

8 de juliol de 2016

Índex

1	Motius	1
2	Definicions	2
3	Per sota de Docker	2
3.1	Sistema de fitxers <i>aufs</i>	2
3.2	<i>Systemd</i>	2
3.3	<i>CGroups</i>	2
3.4	<i>namespaces</i>	2
4	Comandes docker	3
5	Dockerfile	4
6	Volums amb docker	4
7	Mysql Wordpress	5
8	docker-machine	5
9	Conceptes relacionats	5
9.1	Open Container Project	5
9.2	Paravirtualització	5
9.3	Google apps engine	5
10	Sobre aquest document	6

1 Motius

Són múltiples els motius per als quals fem servir Docker. Un dels motius és la portabilitat d'aplicacions i el desplegament d'aplicacions en entorns virtuals molt lleugers (Contenidors Docker) instal·lant en aquests entorns les llibreries i serveis necessaris per a que l'aplicació funcioni i no haver de fer-ho en la teva màquina o en una màquina virtual sencera. Per la mateixa raó, els testing d'aplicacions resulten també molt més lleugers fent servir aquests entorns. D'aquesta manera, doncs, podem penjar contenidors amb l'aplicació i llibreries necessàries per a que s'executin correctament en qualsevol màquina i sense gaire pes.

Hi han també motius de seguretat: les aplicacions queden enjaulades en els contenidors, els quals poden tenir menys funcionalitats i menys comandes del comú.

2 Definicions

i++i

- **Imatge:** Una imatge és un sistema de fitxers i una sèrie de paràmetres de configuració per fer servir en temps d'execució. Mai canvia. Una imatge pot contenir software instal·lat tot el complex com volguem: bases de dades, serveis, etc. Qualsevol pot crear imatges i penjar-les a *Docker Hub*, per exemple, i baixar-te-les per carregar-les a qualsevol contenidor.
- **Contenidor:** Els contenidors es creen a partir d'imatges base. Són instàncies d'imatges. Les imatges es carreguen en el contenidor quan creem una instància d'aquest.
- **Docker Hub:** Repositori d'imatges que la gent penja per compartir. Hi han imatges penjades pels usuaris i imatges oficials (*library*). Pot sincronitzar-se amb *GitHub* de manera que podem tenir un Dockerfile al GitHub per crear imatges.

i++i

3 Per sota de Docker

i++i

3.1 Sistema de fitxers *aufs*

i++i Tipus de sistema de fitxers. Prové de *ufs*, unió de sistemes de fitxers. Per explicar-ho d'una manera fàcil, la manera com funciona aquest tipus de sistemes de fitxers és com la d'un controlador de versions sobre l'arbre de directoris, de tal manera que a mesura que anem fent canvis sobre el sistema es van creant noves versions, de tal manera que podem tirar enrera, utilitzar determinades versions, et.

3.2 *Systemd*

i++i

3.3 *CGroups*

i++i

3.4 *namespaces*

i++i

4 Comandes docker

```
docker run -ti debian /bin/bash
```

Executem el procés `/bin/bash` en un nou contenidor basat en la imatge *debian* obrint terminal interactiu¹. En el moment que acaba la comanda que hem executat en el contenidor amb l'ordre `run` el contenidor s'atura. Si, com en aquest exemple, volem sortir del `bash` que hem executat però sense aturar-ho, només per canviar al terminal nostre, podem fer-ho amb `CTRL+P+CTRL+Q`. Per tornar un altre cop al `/bin/bash` del contenidor fem un

```
docker attach id_contenidor
```

Si el contenidor s'atura (perquè l'hem aturat amb la comanda `docker stop` o perquè la comanda amb la que l'hem iniciat ha terminat) el podem tornar a iniciar amb un `docker start`.

Per eliminar el contenidor, `docker rm ID`. Exemples:

- Arrencar en FG:

```
docker run -ti imatge comanda
```

- veure els contenidors que tenim (aturats o no)

```
docker ps -a
```

- Esborrar tots els contenidors que tenim:

```
docker ps -qa | xargs docker rm -f
```

- Executar una comanda en un contenidor que està iniciat

```
docker exec [-ti] ID CMD
```

- Buscar imatges en *DockerHub*

```
docker search PATRO_DE_BUSQUEDA
```

¹Quan nosaltres fem referència a una imatge com en l'exemple d'abans, el que fa *docker* és buscar la imatge en local, si no la troba la va a buscar a *dockerhub*.

- Veure imatges locals

```
docker images
```

- Veure ip del contenidor

```
docker inspect --format '{{ .NetworkSettings.IPAddress }}' 9
fe0f41200a1
```

- Mapejar ports:

```
# Mapeja automàticament tots els ports
# exposats pel contenidor.
# Assigna ports alts
docker run -P -d kpeiruza/lpic2-apache2-auth
```

- Mapejem port amb port de l'anfitrió

```
# Mapeja el port 80 del contenidor al 8000 de l'amfitrió
docker run -p 8000:80 -d kpeiruza/lpic2-apache2-auth
```

- Llistar els volums creats:

```
docker volume ls
```

i++i

5 Dockerfile

fitxer de configuració per fer el build d'una imatge.

Què podem fer amb un *Dockerfile*? Per exemple, podem executar comandes sobre un contenidor amb el fitxer dockerfile i crear una imatge novai, a més, pujar-la a git i a la vegada enllaçar-ho amb dockerhub amb la comanda

```
docker build -t <etiqueta>
```

el nom de l'etiqueta és e nom que li volem donar a la nova imatge.

6 Volums amb docker

Amb Docker és possible crear volums interns, volums que siguin persistents i que estiguin identificats amb un nom i volums mapejats amb directoris o volums externs.

Per exemple, pot ser interessant crear un volum mapejat amb un directori de la nostra màquina anfitriona de tal manera que en ell hi haguessin allotjades totes les dades d'una base de dades. Així, les dades persistirien encara que el contenidor s'aturès.

Un altre exemple. Exercici:

Crea una imatge binària amb nom i un volum amb nom per a reproduir música. Pots emprar comandes com mpg123 o mplayer. Per on començar? executa

un contenidor amb 2 volums, un de persistent (amb nom) i un que mapegi un directori del sistema a un del contenidor, per tal de passar-nos els mp3 amb la comanda `cp`.

Comanda que crea el contenidor amb els volums que toquen:

```
docker run --device=/dev/snd:/dev/snd --rm -it -v ~/Baixades:/
downloads -v lsmrevamusica:/musica debian /bin/bash
```

Què hem fet? hem iniciat un contenidor executant interactivament una sessió de bash; amb l'opció `-device` podem afegir un dispositiu del host al contenidor. En aquest cas, hem afegit el dispositiu de so²; amb l'opció `-rm` ens assegurem que el contenidor s'esborra quan surt; amb les opcions `-v` creem els volums. El primer volum (`/downloads`) el mapejem amb l'anfitrió (`/Baixades`), després creem un altre volum donant-li l'etiqueta `lsmrevamusica` i que es munta a `/musica`.

Amb això, podríem ara entrar en el contenidor i instal·lar un reproductor mp3 per escoltar música. Com crear ara la imatge? Tenim dues opcions:

1. amb *docker commit*:

```
docker commit 376da04ee184 exerMplayer
```

On el nombre que hi apareix suposo que és l'id del contenidor.

2. creant un *dockerFile*

i++i

7 Mysql Wordpress

En la *url* de la bibliografia [3] hi han els paràmetres necessaris per configurar des de la comanda docker usuaris, passwords i bases de dades. Exemple:

```
docker run --name wordpress -e WORDPRESS_DB_PASSWORD=wordpress -p
8080:80 --link wpdb:mysql -ti wordpress
```

Ara hem de fer persistents les dades: les de mysql i les de wordpress. Com? mapejant-les amb un directori a l'anfitrió. Hem de crear volums que es mapejin amb aquests directoris. El servei apache aleshores aniria en un altre contenidor i hauríem de mapejar el seu `/var/www/html` amb un directori de l'anfitrió.

8 docker-machine

i++i

Amb docker-machine podem crear una màquina virtual que sigui servidor de màquines docker. Amb les comandes de docker podem accedir a la màquina virtual fàcilment, per lo que no cal arrancar-la a ma, per exemple, des del virtualbox. Podem també, amb la comanda *docker-machine* gestionar els contenidors de dintre de la màquina virtual.

² `/dev/snd` és el dispositiu on trobem tots els drivers de totes les targetes de so a Linux

9 Conceptes relacionats

i++i

9.1 Open Container Project

i++i

9.2 Paravirtualització

i++i

9.3 Google apps engine

i++i

10 Sobre aquest document

Copyright© 2016 jaguile apunts.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Referències

- [1] <https://docs.docker.com/linux/>
Getting Started i instal·lació.
- [2] <https://docs.docker.com/engine/installation/linux/debian/>
- [3] https://hub.docker.com/_/mysql/
Com configurar una base de dades en un contenidor.