

Dockerfile - Prima parte

Fabio Ros

Dockerizing / Containerizing

Per creare immagini si utilizza un particolare file chiamato

Dockerfile

Il Dockerfile descrive:

- come deve essere costruita l'immagine che descrive
- la modalità (di default) con la quale verrà avviato un container a partire da essa.

Di default il builder di Docker ricerca il Dockerfile nella root del progetto, tuttavia è possibile specificare un percorso diverso con un apposito flag (-f) nel comando di build.

Documentazione ufficiale:

<https://docs.docker.com/engine/reference/builder>

Esempio che utilizzeremo per comprendere alcuni concetti ricorrenti:

https://hub.docker.com/_/mysql

Nella sezione “Supported tags and respective Dockerfile links” troveremo i link dei Dockerfile.

Prenderemo in considerazione la versione 8.0.

Dockerfile: Sitassi

FROM

Specifica l'immagine base da cui partire per costruirne una di nuova.

Dockerfile: Sintassi > FROM > Esempio

```
FROM alpine:latest
```

ARG

Specifica un parametro da passare al comando FROM.

Dockerfile: Sintassi > ARG > Esempio 1

```
ARG VERSIONE_IMG=latest  
FROM alpine:${VERSIONE_IMG}
```

Dockerfile: Sintassi > ARG > Visibilità (Scope)

I parametri impostati mediante ARG, sono visibili solo al comando FROM perché visibili solo al di fuori della build.
Per poterli utilizzare altrove vanno dichiarati anche dopo il FROM senza assegnargli un nuovo valore

Dockerfile: Sintassi > ARG > Esempio 2

```
ARG VERSION=latest
FROM alpine:${VERSION}
ARG VERSION
RUN echo $VERSION > file.txt
```

RUN

Esegue un comando creando di fatto un nuovo layer

Dockerfile: Sintassi > Modalità di invocazione Bash

- Esegue il comando in una shell
 - Il comando viene fatto precedere dall'invocazione di una shell.
In ambiente unix: `"/bin/sh -c"`
-

Esempio:

```
RUN echo `testo di prova`
```

Dockerfile: Sintassi > Modalità di invocazione Exec

- Compone il comando da eseguire concatenando le stringhe disposte in un array json
- Le stringhe devono essere racchiuse tra doppi apici ed utilizzare \ (backslash) come carattere di escape

Esempi:

```
RUN ["sh", "-c", "echo hello"]
```

```
RUN ["/bin/bash", "-c", "echo hello"]
```

```
RUN ["file_eseguibile.sh"]
```

```
RUN ["sh", "-c", "echo $VARIABLE"]
```

Attenzione: La cache dei layer viene invalidata solo dalle modifiche al Dockerfile e non dalle modifiche esterne ad esso.

Eseguendo il comando di **docker build** con il flag **--no-cache** questo problema non si pone più perché l'immagine viene ricostruita invalidando tutti i layer a partire dalla prima occorrenza del comando RUN presente nel Dockerfile

Concatenazione

Attenzione: "&&" è diverso da ";"

`cmd1 && cmd2` : `cmd2` viene eseguito solo se `cmd1` termina con successo

`cmd1 ; cmd2`: vengono sempre eseguiti entrambi i comandi in ordine di comparizione

Concatenazione > Esempio

```
false ; echo prova    ~> stampa prova  
false && echo prova    ~> nessuna stampa
```

Per ridurre il numero di layer creati contestualmente ai comandi RUN nel Dockerfile è buona pratica accorpare più istruzioni in un unico comando utilizzando la concatenazione.

Per rendere il tutto più leggibile si tende a riportare ogni istruzione su una nuova riga utilizzando il carattere backslash al termine della riga da interrompere

CMD

Specifica l'istruzione che viene eseguita all'avvio dei container generati dall'immagine che stiamo costruendo con il Dockerfile

CMD viene invocato:

- All'avvio di un nuovo container
- Allo start di un container che era stato precedentemente messo in stato di stop

È possibile sovrascrivere questo comando impostandone uno di diverso in coda all'invocazione del comando della CLI:
`docker container run.`

Es. `docker container run alpine:latest sleep 10`

Analogamente al comando RUN, anche CMD può essere specificato sia con la sintassi bash sia con quella exec

Se sono specificate più occorrenze di CMD, sarà presa in considerazione solo quella che compare per ultima nel Dockerfile

Cosa cambia tra RUN e CMD?

Dockerfile: Differenza tra RUN e CMD

RUN :

- Build Time
- Ha effetto sulla costruzione dell'immagine

CMD :

- Run Time
- Ha effetto sui container generati da tale immagine