moku
www.moku.io

Dockerfile - Seconda parte

Fabio Ros

Dockerfile: Sintassi > ENTRYPOINT

ENTRYPOINT

Permette di configurare le istruzioni che deve eseguire il container prima di passare all'esecuzione di quanto specificato da CMD

Dockerfile: Sintassi > ENTRYPOINT

- È possibile sovrascrivere il comportamento di ENTRYPOINT specificando l'override utilizzando il flag --entrypoint al momento dell'esecuzione di `docker container run`
- La sintassi è analoga a quella prevista per CMD
  - È possibile utilizzare sia la sintassi bash che quella exec
  - In generale la sintassi bash ha la precedenza
- Nel Dockerfile deve essere definito almeno uno tra CMD o ENTRYPOINT

Dockerfile: Sintassi > ENTRYPOINT

Cosa cambia tra CMD ed ENTRYPOINT?

CMD viene utilizzato per definire il comando con cui deve iniziare l’esecuzione di un container avviato a partire dall’immagine generata con il dockerfile corrente.

ENTRYPOINT viene utilizzato per eseguire comandi prima dell’esecuzione di CMD. Spesso si tratta di comandi funzionali alla corretta esecuzione del container, da sovrascrivere solo se si conosce approfonditamente il contesto in cui si opera.

Dockerfile: Sintassi > ENTRYPOINT

Attenzione a come interagiscono CMD ed ENTRYPOINT a seconda del formato (Bash o Exec) che si sceglie di usare!

CMD ENTRYPOINT	No ENTRYPOINT	ENTRYPOINT exec_entry p1_entry	ENTRYPOINT ["exec_entry", "p1_entry"]
No CMD	*error, not allowed*	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry
CMD ["exec_cmd", "p1_cmd"]	exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry exec_cmd p1_cmd
CMD ["p1_cmd", "p2_cmd"]	p1_cmd p2_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry p1_cmd p2_cmd
CMD exec_cmd p1_cmd	/bin/sh -c exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd

es. invocazione shell

Dockerfile: Sintassi > ENTRYPOINT

Attenzione a come interagiscono CMD ed ENTRYPOINT a seconda del formato (Bash o Exec) che si sceglie di usare!

CMD ╲ ENTRYPOINT	No ENTRYPOINT	ENTRYPOINT: Bash ENTRYPOINT exec_entry p1_entry	ENTRYPOINT [“exec_entry”, “p1_entry”]
No CMD	*error, not allowed*	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry
CMD [“exec_cmd”, “p1_cmd”]	exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry exec_cmd p1_cmd
CMD [“p1_cmd”, “p2_cmd”]	p1_cmd p2_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry p1_cmd p2_cmd
CMD: Bash **CMD exec_cmd p1_cmd**	/bin/sh -c exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd

Dockerfile: Sintassi > ENTRYPOINT

Attenzione a come interagiscono CMD ed ENTRYPOINT a seconda del formato (Bash o Exec) che si sceglie di usare!

CMD ╲ ENTRYPOINT	**No ENTRYPOINT**	**ENTRYPOINT exec_entry p1_entry**	**ENTRYPOINT [“exec_entry”, “p1_entry”]**
No CMD	*error, not allowed*	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry
CMD [“exec_cmd”, “p1_cmd”]	exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry exec_cmd p1_cmd
CMD [“p1_cmd”, “p2_cmd”]	p1_cmd p2_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry p1_cmd p2_cmd
CMD exec_cmd p1_cmd	/bin/sh -c exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd

Dockerfile: Sintassi > ENTRYPOINT

Attenzione a come interagiscono CMD ed ENTRYPOINT a seconda del formato (Bash o Exec) che si sceglie di usare!

CMD ╲ ENTRYPOINT	No ENTRYPOINT	ENTRYPOINT exec_entry p1_entry	ENTRYPOINT [“exec_entry”, “p1_entry”]
No CMD	*error, not allowed*	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry
CMD [“exec_cmd”, “p1_cmd”]	exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry exec_cmd p1_cmd
CMD [“p1_cmd”, “p2_cmd”]	p1_cmd p2_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry p1_cmd p2_cmd
CMD exec_cmd p1_cmd	/bin/sh -c exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd

Dockerfile: Sintassi > ENTRYPOINT

Attenzione a come interagiscono CMD ed ENTRYPOINT a seconda del formato (Bash o Exec) che si sceglie di usare!

CMD ╲ ENTRYPOINT	No ENTRYPOINT	ENTRYPOINT exec_entry p1_entry	ENTRYPOINT [“exec_entry”, “p1_entry”]
No CMD	*error, not allowed*	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry
CMD [“exec_cmd”, “p1_cmd”]	exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry exec_cmd p1_cmd
CMD [“p1_cmd”, “p2_cmd”]	p1_cmd p2_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry p1_cmd p2_cmd
CMD exec_cmd p1_cmd	/bin/sh -c exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd

Dockerfile: Sintassi > ENTRYPOINT

Attenzione a come interagiscono CMD ed ENTRYPOINT a seconda del formato (Bash o Exec) che si sceglie di usare!

CMD ENTRYPOINT	**No ENTRYPOINT**	**ENTRYPOINT exec_entry p1_entry**	**ENTRYPOINT [“exec_entry”, “p1_entry”]**
No CMD	*error, not allowed*	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry
CMD [“exec_cmd”, “p1_cmd”]	exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry exec_cmd p1_cmd
CMD [“p1_cmd”, “p2_cmd”]	p1_cmd p2_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry p1_cmd p2_cmd
CMD exec_cmd p1_cmd	/bin/sh -c exec_cmd p1_cmd	/bin/sh -c exec_entry p1_entry	exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd