Docker:

Un container è una sorta di scatola che contiene tutto quello che serve per eseguire codice, è come una macchina virtuale ma più efficiente, usa le risorse solo quando gli servono.

Cosa posso fare con docker?

Posso hostare il mio sito web, testare app o ambienti di sviluppo: è un modo veloce ed efficiente per testare o usare app e servizi.

Docker ha due versioni: Community (Gratuita) CE e Enterprise (a pagamento) EE.

Differenza tra container e immagine:

Il container è un ambiente di esecuzione di un'immagine, inoltre il container virtualizza il sistema (come una macchina virtuale) ha un suo file system.

In Docker hub possiamo trovare varie immagini scaricabili nel nostro ambiente con l'apposito comando per eseguire il pull di quell'immagine nel prompt.

docker pull nome\_img -> estrae l'immagine dalle librerie nel file system del container docker

docker run nome\_img -> crea un container per l'immagine

docker ps nome\_img -> da altre informazioni sull'esecuzione dell'immagine con ctrl+c si stoppa l'esecuzione

docker run -d nome\_img -> restituisce in output l'ID del container in cui gira l'applicazione in detach mode

docker stop id\_container -> stoppa quel container

docker start id\_container -> runna quel container

docker ps -a -> mostra tutti i container che stanno runnando o no (una sorta di cronologia)

docker run nome\_img:version -> corrisponde a fare una pull + una run

Nei log, raggiungibili da docker ps abbiamo tutte le info TCP ad esempio e per quanto riguarda le porte coinvolte.

docker run -p6000:6379 -> runno un container sulla porta 6000 del mio pc e sulla 6379 del container (binding porta/container).

Ovviamente se una porta del pc è occupata, non posso runnarci nulla dentro, a differenza di quella al quale è associato il container (es 6379).

docker images -> ci mostra la lista di tutte le immagini che abbiamo scaricato

docker logs container-id -> mostra il logs di quel container

docker run (-d -pxxxx:xxxxx) --name nome\_container nome-img:version -> per dare un nome al container

docker exec -it container\_id /bin/bash -> ottengo un terminale del container (interactive terminal)

Quì dentro potrò fare:

ls

pwd

cambiare directory

ecc. tutto in sintassi linux

exit -> mi fa uscire dal terminale

AL POSTO DELL'ID POSSIAMO ANCHE USARE IL NOME DEL CONTAINER.