

Cloud Computing Lab 2 – Docker

Checklist

*Rispondere alle domande **IN SEQUENZA***

L'output della domanda X dipende dalle precedenti X-1 domande

1. Eseguire il comando `docker run hello-world` e riportare l'output ottenuto:

2. Se si riesegue il comando precedente, la sua esecuzione è più veloce? Se sì, perché?

- 3. Spiegare cosa fa il seguente comando, compresi i flags:**

```
docker run -i -t debian /bin/bash
```

- 4. Dopo aver eseguito il comando precedente, cosa succede eseguendo il comando `exit`? Perché?**

- 5. Eseguire il comando che permette di ottenere la lista delle immagini al momento presenti sulla propria macchina:**

6. Eseguire il comando che permette di ottenere la lista sia dei container in esecuzione che di quelli eseguiti:

7. Qual è la differenza in un Dockerfile tra i comandi RUN e CMD?

8. Scaricare da moodle il file DockerApp.zip, estrarlo ed eseguire in locale i seguenti comandi all'interno della cartella per lanciare l'applicazione:

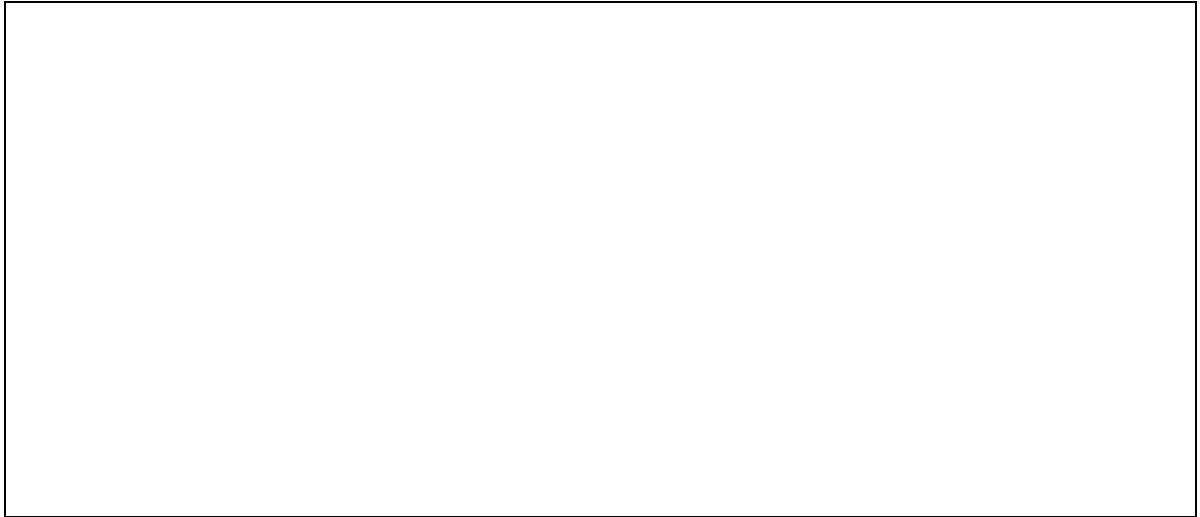
```
a. pip install -r requirements.txt  
b. python3 app.py
```

Andare quindi su <http://127.0.0.1:5000/> e verificare che l'applicazione sia in esecuzione. Dopo averla chiusa containerizzarla ed eseguirla su Docker. Descrivere tutti gli step ed i comandi necessari e gli eventuali flag utilizzati. Verificare che anche l'applicazione containerizzata sia disponibile su <http://127.0.0.1:5000/>

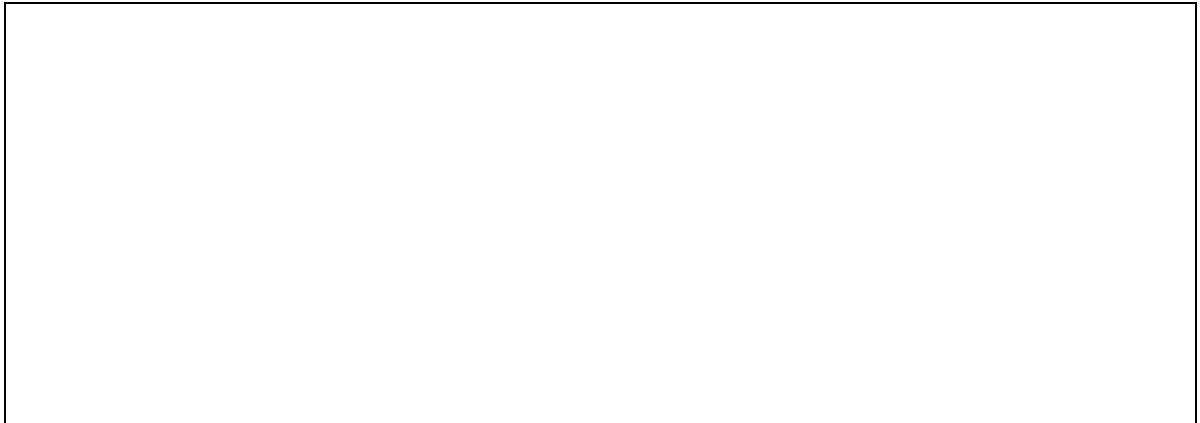
Suggerimento 1: come immagine di partenza usare `python:3.8-slim-buster`

Suggerimento 2: per lanciare la nostra applicazione in un container è necessario eseguire il comando:

```
python3 -m flask run -host=0.0.0.0
```



9. Ridigitare il comando del passo 5 e verificare cosa è cambiato



10. Seguire la seguente guida per provare `docker-compose` :

`https://docs.docker.com/compose/gettingstarted/`