

## Attività 4.c: microservizi – applicazione web

### Esercitazione

**Obiettivo:** avviare un microservizio in un container che fornisca un'applicazione web, con REST API acceduta, per semplicità, lato server

#### Fase 1: prova dell'applicazione sulla macchina host

1. Per poter provare l'applicazione prima di creare l'immagine è necessario aver installato nella macchina host *Python*, *flask* e *requests*. (attenzione: nessun container deve essere attivo)
2. L'applicazione usa i *templates* di Flask: fare riferimento all'esercitazione dello scorso anno
3. Creare, nella cartella *muscolin*, una sotto cartella *web* e in questa una sottocartella *templates*.
4. Nella cartella *web* copiare i file *webApp* e nella cartella *templates* i template HTML (il tutto disponibile zippato allegato all'attività).

```
from flask import Flask,render_template
import requests

app=Flask(__name__)

@app.route("/")
def index():
    return(render_template("index.html"))

@app.route("/corsi")
def corsi():
    elencoCorsi=requests.get("http://xxx.xxx.xxx.xxx:5000/corsi").json()
    return(render_template("corsi.html",elencoCorsi=elencoCorsi))

@app.route("/sedi")
def sedi():
    return(render_template("sedi.html"))    return(risposta)
```

5. Provare l'applicazione sulla macchina host, accedendo da un browser esterno o con cu. (ricordarsi di impostare l'opzione *-p 8000*: la porta 5000 è già in uso dal container *api*)

#### Fase 2: generazione dell'immagine

1. Generare l'immagine *web* (basata sull'immagine *python*), specificando, se necessario, il proxy (vedi attività 3.d) ed installando nell'immagine *flask* e *requests*

```
from python
run python3 -m pip install flask
run python3 -m pip install requests
copy . /
cmd python3 -m flask --app webApp.py run --host 0.0.0.0 --port 8000
```

2. Avviare il container (con nome *web*) e mappare la porta 8000 sulla porta 80

```
$ docker container run --name web --rm -d -p 80:5000 web
```

3. Provare da un browser; in caso di problemi lanciare i container in modalità interattiva e verificare i messaggi