

Progetto DSBD Diego Calabretta 1000012346

Il progetto consiste in un sistema di gestione di risultati sportivi e contiene i seguenti microservizi:

- **Event**

È il microservizio di business che gestisce eventi sportivi, entità caratterizzata da: squadraCasa, squadraOspite, punteggioCasa, punteggioOspite, data. Il microservizio contiene il controller con le seguenti API che interagiscono con il database:

- 1) GET event/all : per accedere a tutti gli eventi
- 2) GET event/{id}: accedere ad un evento per id
- 3) GET event/squadra/{squadra}: accedere ad un evento per squadra, che sia squadraOspite o squadraCasa
- 4) POST event/add: aggiungere evento
- 5) DELETE event/{id}: eliminare evento per id

Event è anche un producer di log con timestamp, ip, metodo, status; il log viene generato ad ogni richiesta ed inviato tramite Kafka al topic logging.

- **Consumer logging**

È un microservizio consumer Kafka sottoscritto al topic logging. Una volta ricevuto il log prodotto da Event, esso viene memorizzato sul database.

- **API gateway**

Il microservizio implementa il pattern API gateway instradando il servizio Event dalla porta "8080" alla porta del gateway "9000" con path "/event/**".

- **Mysqldb**

Microservizio per gestire il database "Sport", esso conterrà le tabelle "event" e "log", le entità create rispettivamente dal servizio Event e da Consumer logging. I parametri del database sono presenti nel file ".env"

- **Zookeeper, Kafka**

Microservizi per gestire il broker e i messaggi Kafka.

Docker

I microservizi vengono inseriti in containers tramite il file “docker-compose.yml”. Pertanto, è possibile fare la build dei containers ed eseguirli tramite i comandi: `docker-compose build`, `docker-compose up`

Kubernetes

Il deployment dei microservizi è gestito tramite il file “deployment.yml”. È possibile eseguire il deployment con Minikube con il comando:

```
kubectl apply -f deployment.yml
```

Dopo il deployment potrebbe essere necessario settare l’interfaccia docker in promiscuous mode, in modo tale da accettare self-request, con i comandi:

```
minikube ssh
```

```
sudo ip link set docker0 promisc on
```

Analisi sulle serie temporali

Le analisi sulle serie temporali sono effettuate in un notebook python “TS_Analysis_ARIMA.ipynb” contenuto all’interno della cartella “Time Series Analysis”. Col notebook è possibile fare un’analisi auto ARIMA, il dataset da utilizzare per l’analisi può essere creato con lo script “CSV_creator.py”, esso attinge dal database dei log e crea un file .CSV con numero di richieste ricevuto per giorno. Il file creato verrà nominato “Stats.csv”, tuttavia il vero file utilizzato nell’ analisi è “Stats_test.csv”, un file creato appositamente con lo stesso formato del primo, ma con un numero sufficiente di campioni per allenare un modello.