

Progetto DSBD – TASSI_LENKO

1. Prima parte alternativa

Invece di avere il servizio Producer, Consumer e Handle Notifier (che avevamo inizialmente impostato nella prima parte del progetto), potremmo implementare i seguenti servizi.

Microservizi:

1. **Weather Event Notifier:**

- Questo microservizio si occupa di gestire gli utenti (ex: potremmo registrare l'utente tramite il bot telegram, così otteniamo anche il chat_id da salvare nel database e recuperare nel servizio notification-server) e le loro sottoscrizioni agli eventi meteorologici.
- Espone API REST per la gestione degli utenti, delle città e delle condizioni meteorologiche di sottoscrizione.

2. **Weather Data Fetcher:**

- Periodicamente questo microservizio recupera informazioni meteorologiche tramite l'API di OpenWeatherMap per tutte le città di interesse.
- Processa le informazioni ricevute e le invia al servizio Weather Event Notifier per la verifica delle condizioni.

3. **Notification Service:**

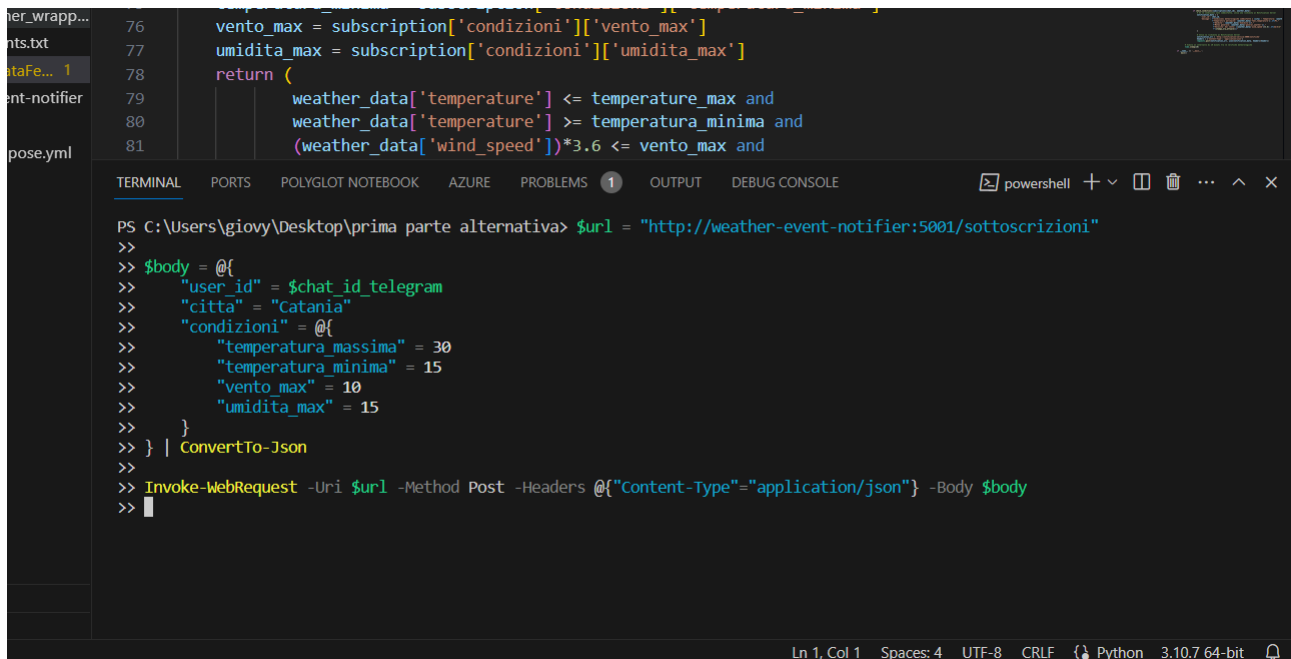
- Questo microservizio si occupa dell'invio delle notifiche agli utenti quando si verificano le condizioni specificate.
- Utilizziamo sempre la libreria per interagire col telegram bot per l'invio delle notifiche.

Abbiamo anche inserito un servizio database-service che si occupa direttamente di interagire col database in modo tale che gli altri 3 servizi non debbano preoccuparsi di connettersi al db o altro.

A differenza della prima parte con questo non utilizziamo KAFKA per comunicare (e su questo abbiamo il dubbio se e dove potrebbe essere necessario implementarlo in questa “prima parte alternativa” – cioè tra quali servizi, per esempio) ma direttamente comunichiamo tramite richieste http agli endpoint esposti.

Sostanzialmente fin ora abbiamo testato il codice avviando docker-compose e poi effettuando la richiesta di una nuova sottoscrizione da parte di un utente.

Esempio richiesta POST di una nuova sottoscrizione da parte dell'utente, sull'endpoint /sottoscrizioni esposto dal servizio WeatherEventNotifier:



```
76     vento_max = subscription['condizioni']['vento_max']
77     umidita_max = subscription['condizioni']['umidita_max']
78     return (
79         weather_data['temperature'] <= temperature_max and
80         weather_data['temperature'] >= temperatura_minima and
81         (weather_data['wind_speed'])*3.6 <= vento_max and

TERMINAL    PORTS    POLYGLOT NOTEBOOK    AZURE    PROBLEMS    1    OUTPUT    DEBUG CONSOLE

PS C:\Users\giovy\Desktop\prima parte alternativa> $url = "http://weather-event-notifier:5001/sottoscrizioni"
>>
>> $body = @{
>>     "user_id" = $chat_id_telegram
>>     "citta" = "Catania"
>>     "condizioni" = @{
>>         "temperatura_massima" = 30
>>         "temperatura_minima" = 15
>>         "vento_max" = 10
>>         "umidita_max" = 15
>>     }
>> } | ConvertTo-Json
>>
>> Invoke-WebRequest -Uri $url -Method Post -Headers @{"Content-Type"="application/json"} -Body $body
>>
```

L'idea, che ancora dobbiamo implementare, potrebbe essere quella di: aggiungere nel servizio weather-event-notifier una gestione della registrazione degli utenti fatta tramite chat col bot telegram eseguendo il comando /start; inserire in una tabella Postgres degli utenti il nomeutente e il chat_id associato all'utente e poi, quindi, inserire dentro la richiesta di una nuova sottoscrizione (figura sopra) il nome utente, al posto del chat_id, che così non dovremmo inserire direttamente nella richiesta e non passerebbe in chiaro (ottenendo un minimo di sicurezza), ma recupereremmo dal db.

Per il resto dobbiamo migliorare anche la logica dei 3 servizi ma la base è ciò che abbiamo fatto fin ora e da cui partire, considerando che comunque non potremo ampliarla troppo poiché dobbiamo occuparci almeno di iniziare la seconda parte e inserire qualcosa (cioè almeno inseriamo Prometheus per il monitoraggio di qualche metrica da testare, il pattern SAGA se riusciamo e poi un servizio per lo SLA Manager e la violazione delle metriche). Sicuramente non riusciremo a fare la parte di Kubernetes e degli algoritmi predittivi per prevedere la violazione di eventuali metriche, però almeno possiamo provare a fare, della seconda fase del progetto, qualcosa (come ho detto sopra tra parentesi).