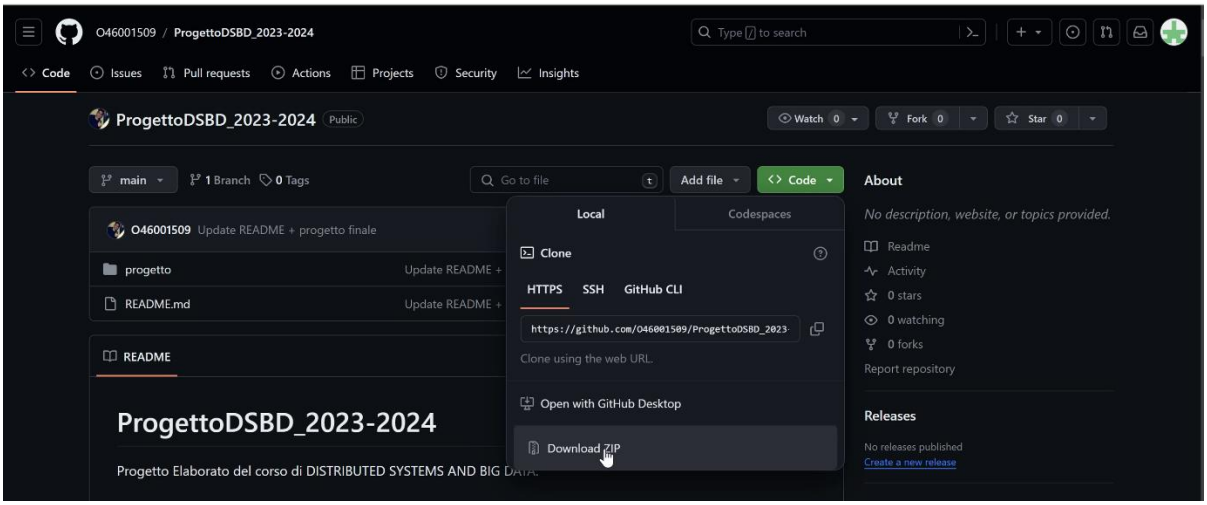
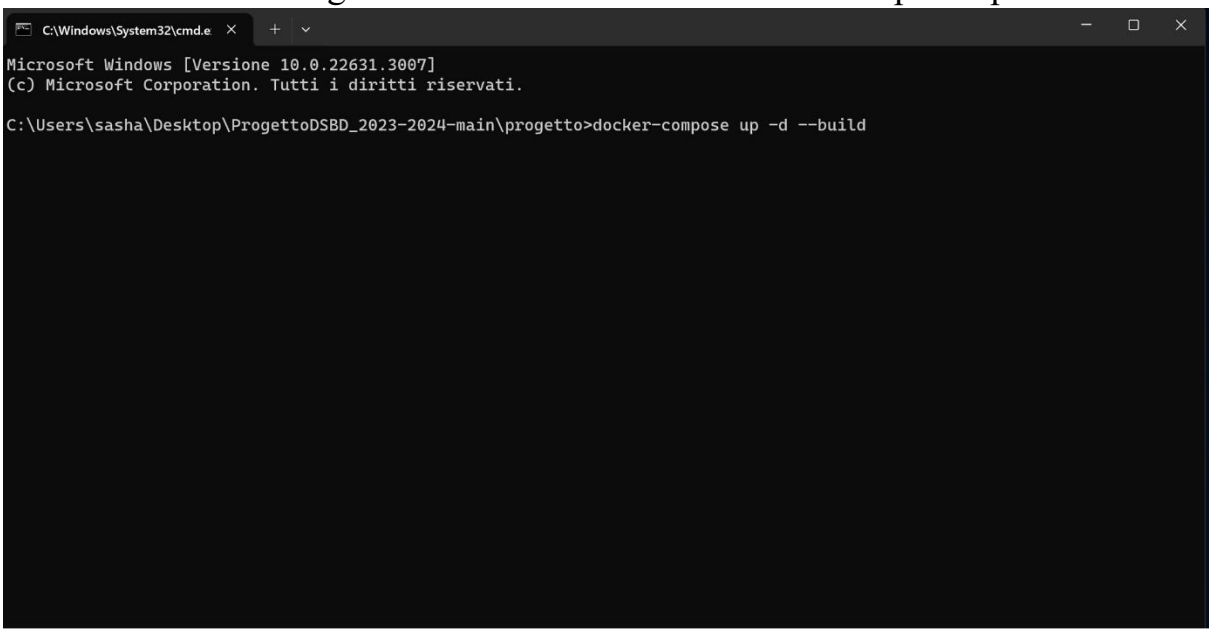
Abbiamo testato il sistema eseguendo i seguenti passi:

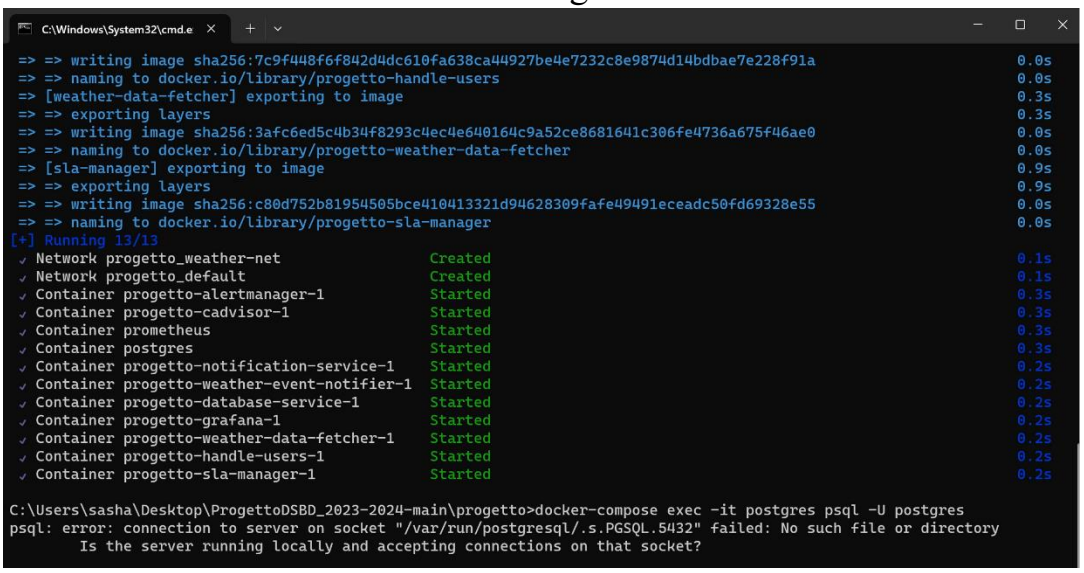
1. Scaricare il file zip da GitHub:



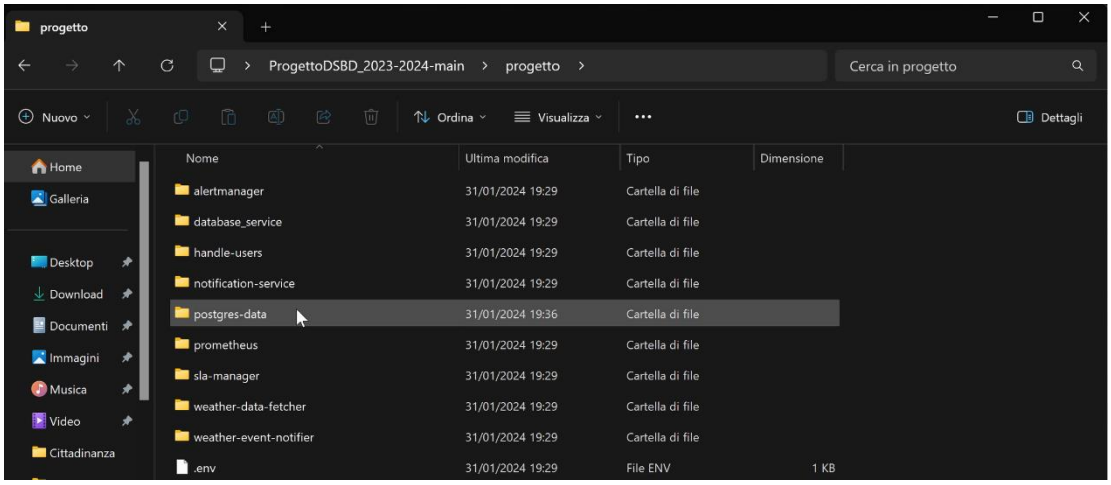
2. Tramite terminale eseguiamo il commando →docker—compose up -d –build



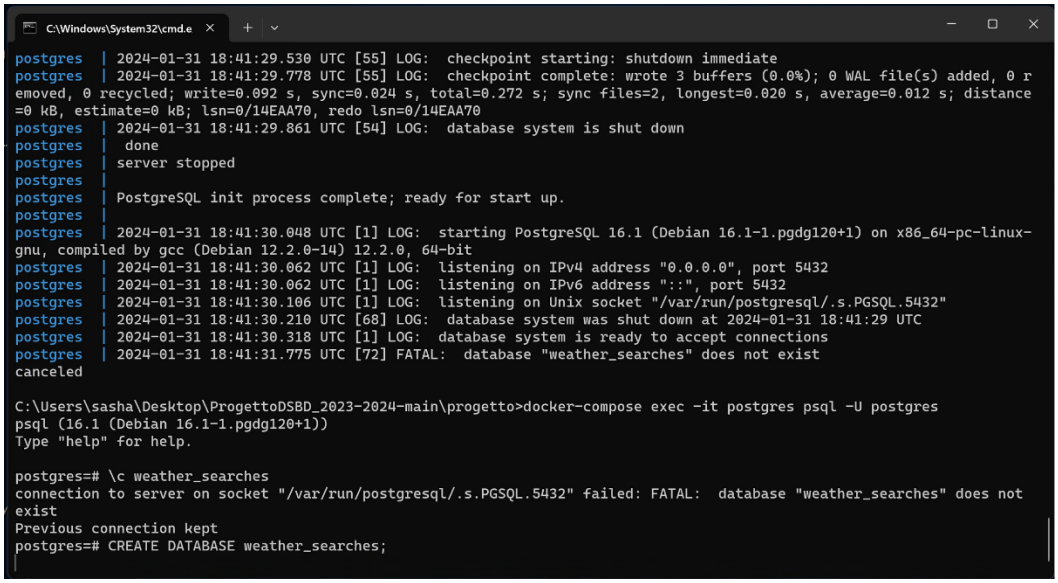
3. Nel caso di un errore come nell'immagine:



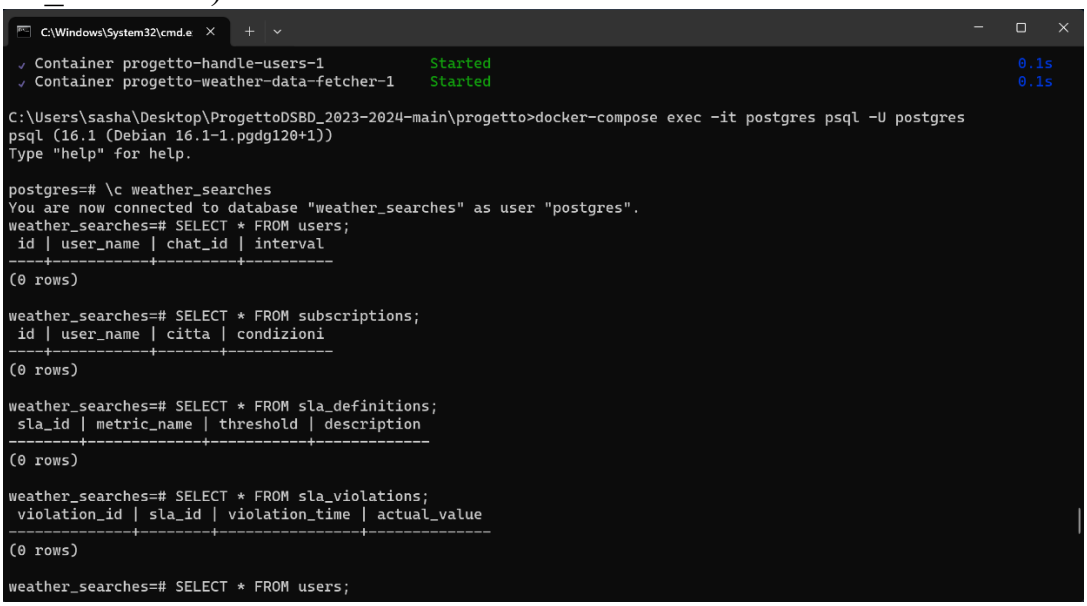
Bisogna eliminare il contenuto della cartella postgres che si trova dentro il progetto:



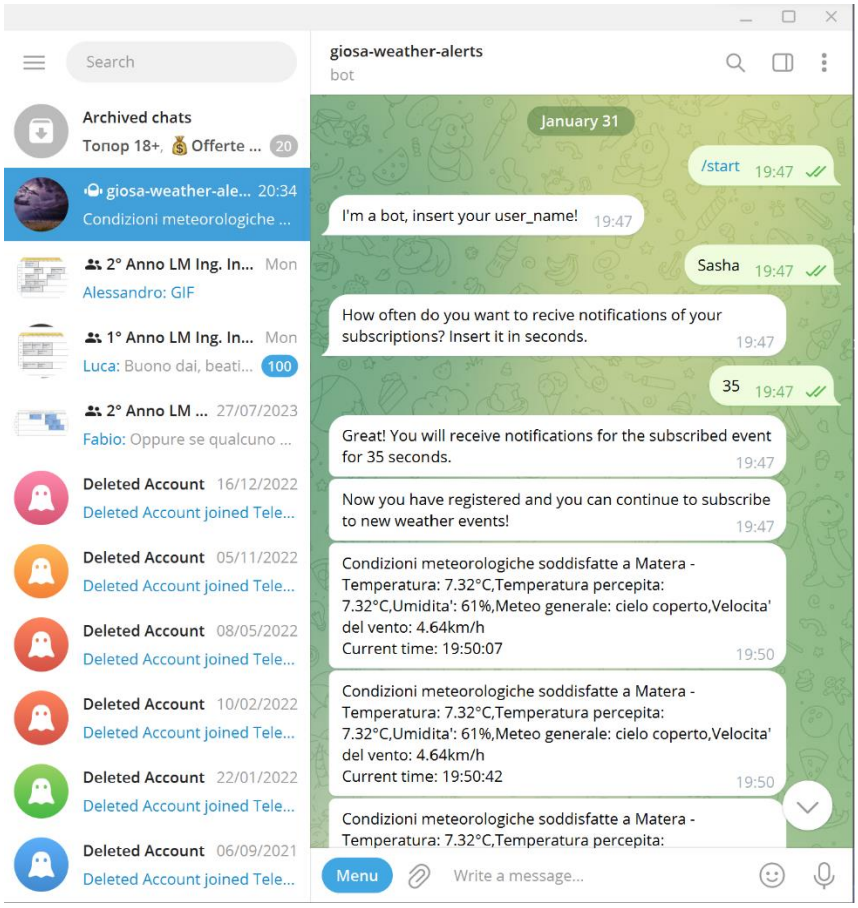
Dopo aver eliminato il contenuto della cartella postgres e riavviato il progetto, creare il database (weather_searches) come nella seguente immagine:



4. Visualizziamo le seguenti tabelle (users, subscriptions, sla_definitions, sla_violations) dentro il nostro database:



5. Nel telegram cerchiamo il bot (giosa-weather-alerts) e cominciamo la conversazione con lui eseguendo la registrazione come mostrato nella seguente immagine:



Per il test abbiamo supposto che l'utente scelga un intervallo notifiche pari a 35 secondi (si può scegliere più grande, ex: 1 minuto, 5 minuti, ecc.) per velocizzare il test dell'applicazione (per esempio non dover aspettare troppo per vedere le violazioni).

6. Creazione della metrica notification_interval_second, inserimento di una nuova sottoscrizione e aggiornamento della sottoscrizione inserita:

```
                                                $url = "http://localhost:5005/sla"
>> $headers = @{"Content-Type" = "application/json"}
>>
>> $body = @{
>>         'metric_name' = 'notification_interval_seconds'
>>         'threshold' = 10
>>         'description' = 'La differenza tra l intervallo effettivo delle notifiche e l intervallo previsto non deve superare il 10% '
>>
>> } | ConvertTo-Json
>>
>> Invoke-WebRequest -Uri $url -Method Post -Headers $headers -Body $body
```

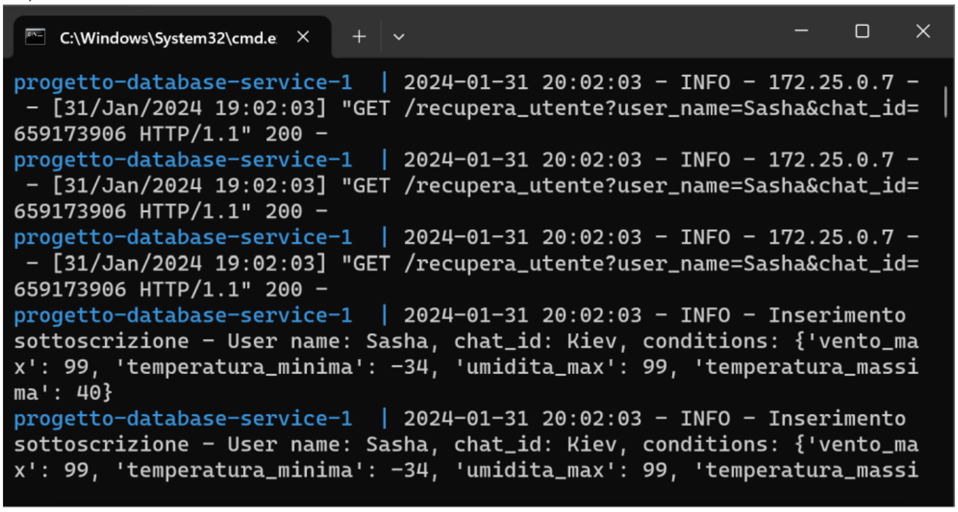
La metrica inserita ha una soglia di 10. Questo significa che si verificherà una violazione nel caso in cui viene superato del 10% l'intervallo notifiche scelto dall'utente.

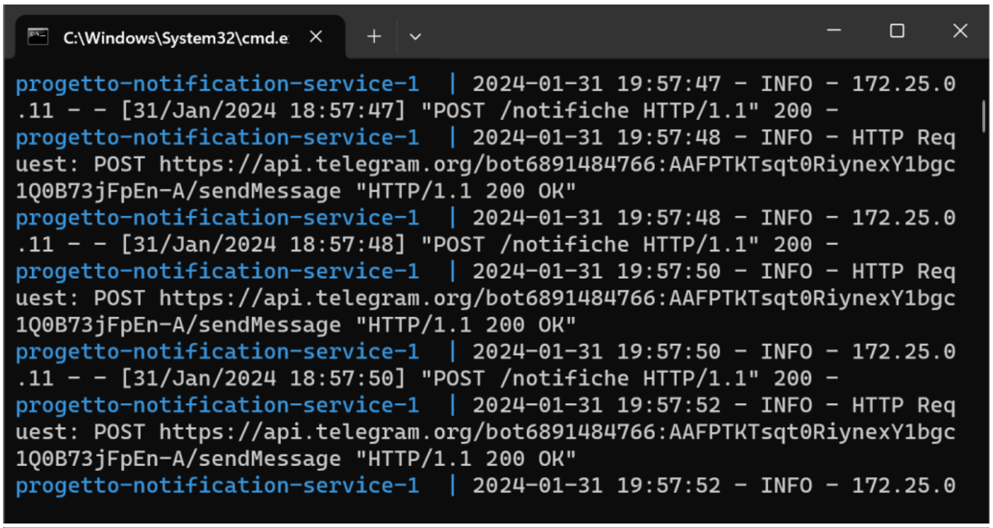
```
                                                $url = "http://localhost:5001/sottoscrizioni"
>> $headers = @{"Content-Type" = "application/json"}
>>
>> $body = @{
>>     "user_name" = "Sasha"
>>     "citta" = "Matera"
>>     "condizioni" = @{
>>         "temperatura_massima" = 40
>>         "temperatura_minima" = -34
>>         "vento_max" = 99
>>         "umidita_max" = 99
>>     }
>> } | ConvertTo-Json
>>
>> Invoke-WebRequest -Uri $url -Method Post -Headers $headers -Body $body
```

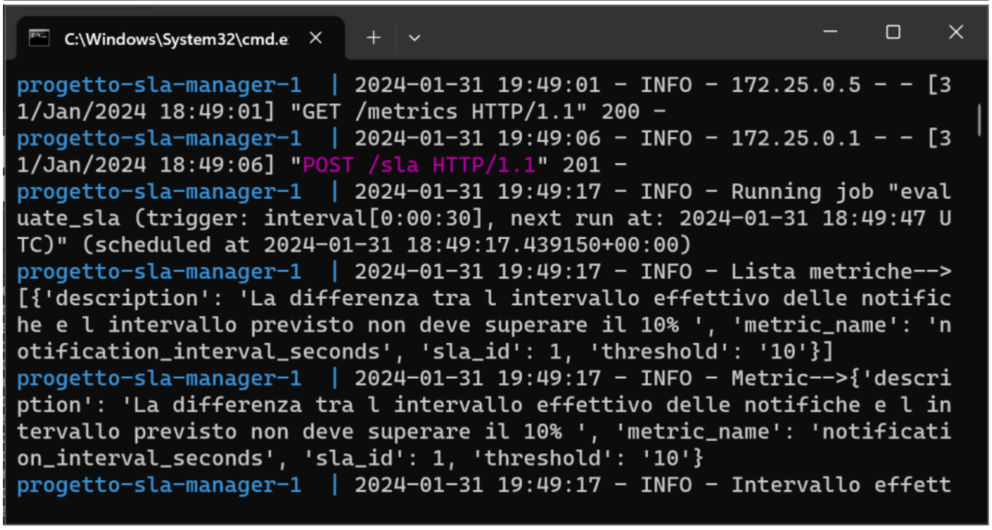
Anche qui per fare in modo di non dover aspettare il verificarsi delle condizioni della sottoscrizione inserita, abbiamo inserito: un range abbastanza grande per la temperatura, e un valore grande per il vento e per l'umidità. In un contesto reale l'utente potrebbe inserire le condizioni che vuole.

```
                                                $url = "http://localhost:5001/sottoscrizioni"
>> $headers = @{"Content-Type" = "application/json"}
>>
>> $body = @{
>>     "user_name" = "Sasha"
>>     "citta" = "Crotone"
>>     "condizioni" = @{
>>         "temperatura_massima" = 20
>>         "temperatura_minima" = 9
>>         "vento_max" = 9
>>         "umidita_max" = 20
>>     }
>> } | ConvertTo-Json
>>
>> Invoke-WebRequest -Uri $url -Method Put -Headers $headers -Body $body
```

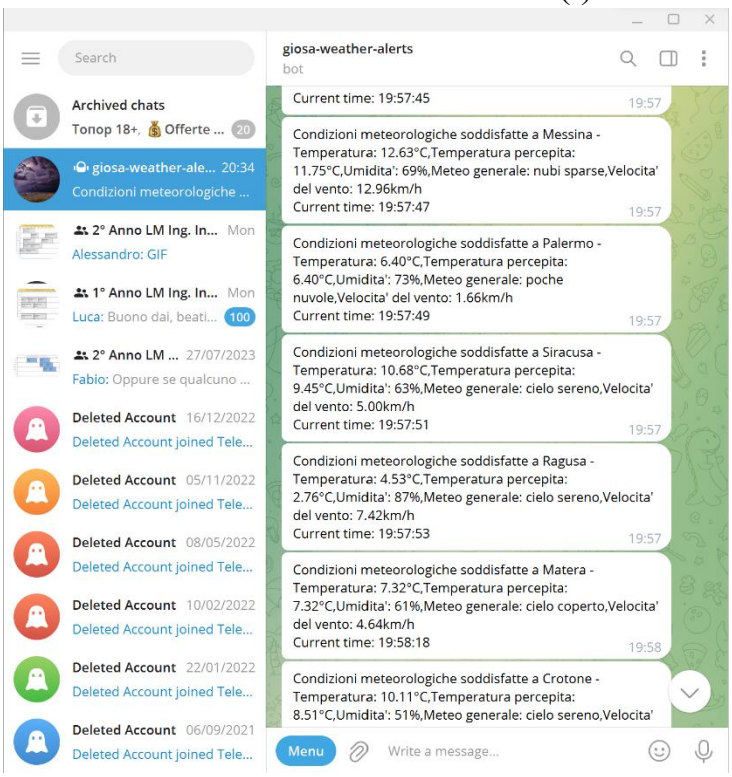
7. Visualizzare logs dei servizi: (docker-compose logs database-service -f, docker-compose logs notification-service -f, docker-compose logs sla-manager -f):



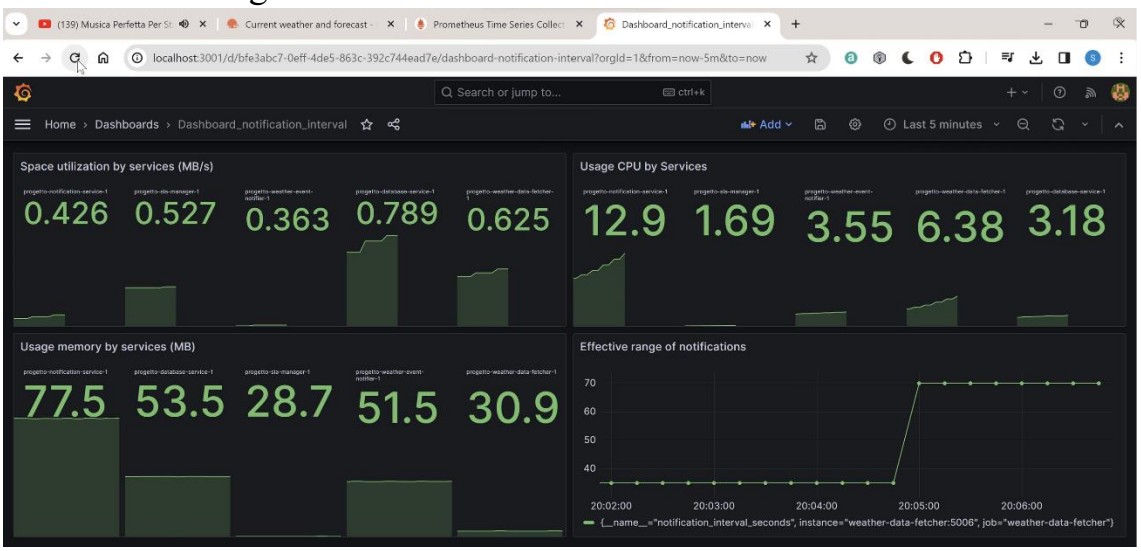




8. Visualizzazione del meteo desiderato(i):

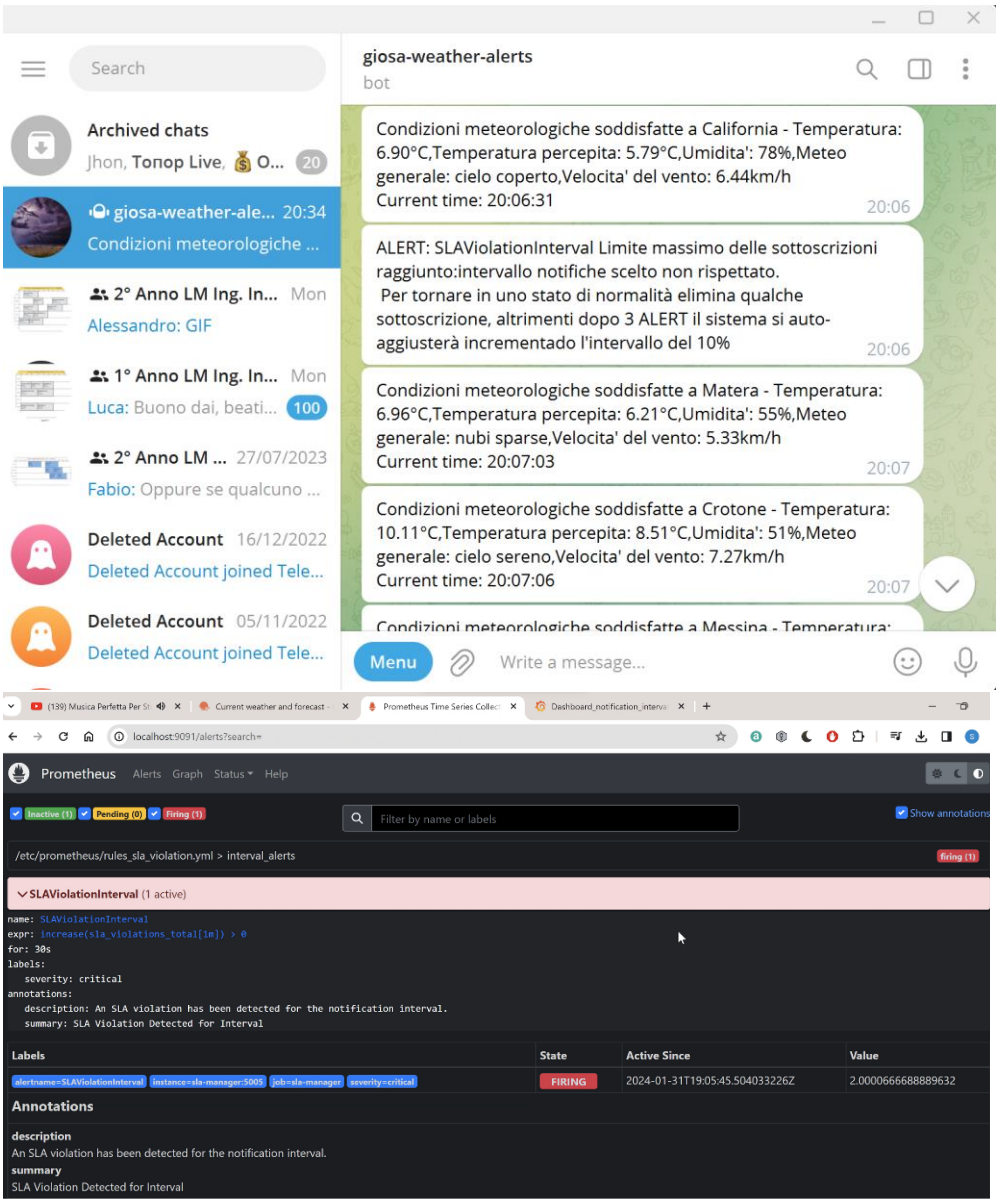


9. Visualizzazione grafici Grafana:



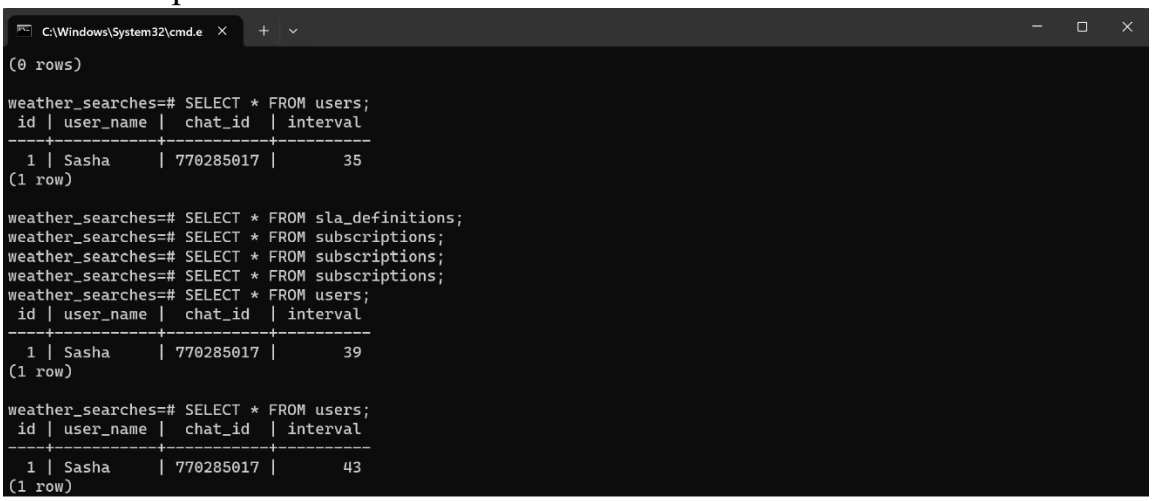
In basso a destra, si vedono, nel grafico del range effettivo di notifiche, dei valori che sono maggiori dell'intervallo scelto dall'utente, secondo la soglia scelta nella metrica (10%). Quindi: intervallo_effettivo > 10% di intervallo notifiche Ex: 70 > 39, dove 39=35+(10% di 35)

10. Nelle seguenti foto si vede l'Alert riferito alla metrica notification_inerval_seconds, che viene inviato dal notification-service all'utente:



11. Come potrebbe eliminare le sottoscrizioni l'utente:

12. Nella seguente immagine si vede il tempo aggiornato dell'arrivo delle notifiche da telegram all'utente dopo l'esecuzione di un'operazione correttiva (se l'utente non ha ancora eliminato alcune sottoscrizioni dopo che gli sono arrivati 3 Alerts), che nel nostro caso mirra ad aumentare l'intervallo notifiche dell'utente per fare rientrare tutte le sottoscrizioni scelte:



13. Nelle seguenti immagini si vede l'inserimento di una nuova metrica usage_memory_by_service e l'eliminazione

14. Nelle seguenti immagini si vede l'Alert riguardo (usage_memory_by_service)