**Immagine che contiene testo, Carattere, schermata, Elementi grafici

Descrizione generata automaticamente**

**Corso di Distribuited Systems and Big Data**

Lorenzo Basile 1000055691, Antonio Santo Buzzone 100055698

**Relazione Progetto: Weather Event Notifier**

1. **Introduzione**

Lo scopo dell’elaborato è lo sviluppo di una piattaforma che permette all’utente di essere avvisato tramite notifica telegram, ad intervalli regolari, su particolari informazioni meteorologiche. L’utente dopo essersi autenticato ha disposizione una funzione di ricerca che gli consente di trovare la città tramite il nome; prima di effettuare la ricerca è importante che l’utente inserisca i propri vincoli sul meteo di cui vuole essere aggiornato.

Una volta trovata la città verranno confrontati i vincoli inseriti dall’utente con le informazioni in tempo reale della città desiderata e successivamente verrà inviata una notifica all’utente.

1. **Architettura dell’applicazione**

**API**

**Gateway**

**Kafkaa**

**ApiService**

**NotifyService**

**AuthService**

**CityService**

**Immagine che contiene Elementi grafici, design, arte

Descrizione generata automaticamente**

**Immagine che contiene Elementi grafici, design, arte

Descrizione generata automaticamenteImmagine che contiene Elementi grafici, design, arte

Descrizione generata automaticamente**

**Kafkaa**

**Prometheus**

*Per il monitoraggio*

L’architettura è composta dai seguenti microservizi:

* AuthService
* CityService
* ApiService
* NotifyService
* Prometheus
* Sla\_manager
* Gateway

AuthService è il microservizio che si occupa dell’autenticazione dell’utente. La sua funzione principale è quella d’interfacciarsi col database MySQL. Se l’utente non risulta essere registrato allora è possibile farlo inserendo alcuni dati come (username, telegram chat id, password), le informazioni dell’utente verranno inserite all’interno della tabella *user* e all’utente verrà assegnato un id univoco.Altrimenti se risulta essere registrato è possibile effettuare il login tramite username e password. Nella fase di registrazione il telegram chat id risulta essere essenziale in quanto all’utente verranno inviate delle notifiche tramite app telegram. È possibile ottenere questo il telegram chat id attraverso la guida nel file readme.

Il secondo microservizio è CityService, dopo avere ricevuto i dati dall’utente tra cui la città che si desidera cercare e le varie constraint, il suo compito è quello d’inserire questi dati e il corrispettivo id dell’utente all’interno di una tabella chiamata info\_meteo. I metdo principali di questo microservizio sono:

* home()
* send\_Kafka()

ApiService è il microservizio che, dopo esser stato avviato interroga costantemente il database per recuperare i dati da info\_meteo, in particolare la città di cui si vogliono ricevere informazioni, tramite un servizio web che mette a disposizione delle Api che consentono di ottenere molteplici informazioni metereologiche, effettua un confronto tra i vincoli inseriti dall’utente le info recuperate dal servizio web. Successivamente di occupa di pubblicare nel topic Kafka “WeatherNotification” queste informazioni tramite un evento. Questi eventi saranno consumati dal microservizio NotifyService. I metodi principali sono:

* consuma\_da\_Kafka()
* checkWeather()
* send\_Kafka()

NotifyService è l’ultimo microservizio progettato per questa piattaforma, che ha come obiettivo primario quello d’inviare una notifica all’utente. Attraverso la ricezione di un evento tramite kafka estrae le informazioni, tra cui id\_utente, telegram chat id, città e vari vincoli e si occupa di inviare tramite il telegram chat id una notifica all’utente. I metodi principali sono:

* consuma\_da\_kafka()
* send\_t\_message()

Infine, l’elaborato contiene anche l’utilizzo di Prometheus: si tratta di un tool di monitoraggio che archivia metriche in un database proprietario di timeseries, dove una timeseries (serie temporale) non è altro che una serie di data point indicizzati in ordine temporale. In particolare, la scelta è stata è stata orientata verso il blackbox monitoring che consiste nel monitorare le risorse computazionali (es. CPU, ram) utilizzate dagli ambienti di esecuzione.

1. **Descrizione dei servizi del sistema distribuito**

Come IDE per lo sviluppo del codice e per l’organizzazione dei package è stato utilizzato intellij, in particolare Python con l’uso di flask.

**3.1 AuthService**

Il microservizio è composto dai seguenti metodi:

* user\_register(): controlla la validità dei campi inseriti, s’interfaccia col DB per l’inserimento del nuovo utente ed infine genera un token che verrà codificato, contenente: id utente e telegram chat id.
* user\_login(): controlla la validità dei campi inseriti e se corretti genera un token analogo al caso precedente