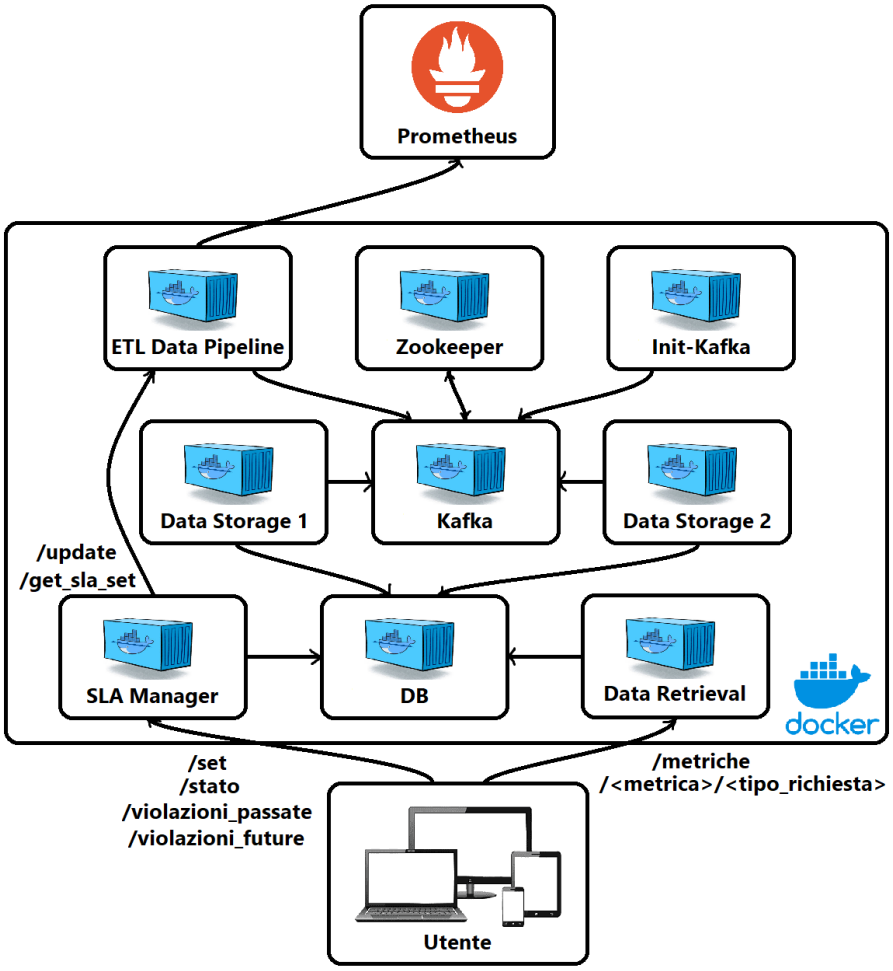
Relazione progetto DSBD

Agata Messina – 1000040309

Firhaan Mohammad Mohun – 1000040311

Sommario

Introduzione e architettura del sistema



Si vuole implementare un sistema di elaborazione dei dati la cui architettura è rappresentata in figura. I dati sono forniti al sistema da un server Prometheus esterno che espone le metriche ottenute dal monitoraggio di una applicazione. Il sistema, composto dai servizi deployati nei container in figura, esegue quindi le seguenti elaborazioni:

1. Il servizio ETL Data Pipeline, ottenuti i dati sulle metriche da Prometheus, esegue delle elaborazioni e invia i risultati ad un broker Kafka nel container kafka;
2. I servizi Data Storage consumano i messaggi dal broker Kafka e li memorizzano in un database contenuto nel container db;
3. Il servizio Data Retrieval, a seguito delle richieste REST fatte dall'utente, ritorna i dati di interesse contenuti nel database;
4. Il servizio SLA Manager, a seguito delle richieste REST fatte dall'utente, permette di cambiare le elaborazioni fatte sulle metriche dal servizio ETL Data Pipeline e di ritornare altri dati di interesse contenuti nel database.

Ogni servizio è deployato all'interno di un container Docker. I vari container possono essere avviati automaticamente mediante un docker compose secondo l'ordine stabilito dai tag depends_on specificati. Le scelte fatte in questo senso sono spiegate nei successivi paragrafi dei relativi servizi. Tutti i container sono inseriti in una network custom "dsbd" al fine di permettere la comunicazione tra di essi specificando il solo hostname e sono configurati per riavviarsi in caso di crash attraverso il tag restart: always.

Componenti principali del sistema

ETL Data Pipeline

Il servizio ETL Data Pipeline si occupa di:

1. Caricare da un file "SLA.txt" i valori di SLA Set e relativi SLO di default.
2. Ottenere i dati di un set di 8 metriche da Prometheus per gli ultimi 7 giorni o per le ultime 1, 3 o 12 ore ogni volta che è necessario. Le richieste a Prometheus vengono ritentate finché si verificano errori.
3. Calcolare ogni 7 giorni i valori di stazionarietà, stagionalità e autocorrelazione per il set di 8 metriche considerando i campioni degli ultimi 7 giorni. La stazionarietà è stata calcolata usando l'Augmented Dickey-Fuller Test; la stagionalità è stata calcolata effettuando la trasformata discreta di Fourier della serie temporale e invertendo la frequenza corrispondente al valore massimo; per l'autocorrelazione si sono memorizzati i lag per cui la serie è autocorrelata (coefficiente di autocorrelazione maggiore di 0.5).
4. Calcolare ogni 3 minuti i valori di massimo, minimo, media e deviazione standard per le stesse metriche del punto precedente considerando i campioni delle ultime 1, 3 e 12 ore.
5. Predire ogni 3 minuti i valori di massimo, minimo e media per un sottoinsieme di 5 metriche (SLA Set) nei successivi 10 minuti.
6. Calcolare ogni 3 minuti il numero di violazioni per l'SLA Set nelle ultime 1, 3 e 12 ore.
7. Predire ogni 3 minuti eventuali violazioni future dell'SLA Set considerando i campioni delle ultime 1, 3 e 12 ore.
8. Esporre 2 endpoint per impostare (/update) e ottenere (/get_sla_set) l'SLA Set e i relativi SLO (utilizzati dal servizio SLA Manager). Quando viene impostato un nuovo SLA Set con i relativi SLO tramite /update, tali informazioni vengono anche salvate nel file "SLA.txt" per fornire persistenza.
9. Inviare i dati calcolati nei punti 3, 4, 5, 6 e 7 al servizio Kafka nel topic "prometheusdata". L'invio viene ritentato finché si ottengono errori retriable.
10. Creare e aggiornare un file di log ("log.txt") contenente i tempi di esecuzione delle elaborazioni dei punti 4, 5, 6 e 7. Il file aggiornato dal servizio nel container coincide con quello presente nella cartella del progetto, all'interno della directory "ETL_Data_Pipeline", in quanto nel docker compose si è fatto il bind mount.

Il servizio ETL Data Pipeline è deployato in un container chiamato "etl_data_pipeline". Tale container espone la porta 50000, come specificato nel dockerfile, affinché i suoi endpoint possano essere raggiunti dal servizio SLA Manager. Il container viene avviato successivamente ai due container di Data Storage affinché ci siano già dei consumer Kafka pronti a ricevere i messaggi prodotti.

Data Storage

Il servizio Data Storage si occupa di:

1. Creare un consumer che si sottoscrive al topic "prometheusdata" del servizio Kafka. Se si ottengono errori retriable si ritenta la sottoscrizione.
2. Connettersi al database del servizio db. Se la connessione fallisce ritenta ogni 5 secondi.
3. Fare il polling continuo dei nuovi messaggi dal servizio Kafka.
4. Memorizzare il contenuto dei messaggi ricevuti nel servizio db. L'inserimento di dati in più tabelle avviene tramite transazioni.

All'interno dell'applicazione sono previsti 2 servizi Data Storage deployati in due container chiamati "data_storage_1" e "data_storage_2". I due container, che implementano due consumer membri dello stesso consumer group in quanto configurati con lo stesso parametro group.id, vengono avviati dopo la chiusura del container init_kafka così da poter operare quando il topic

“prometheusdata” nel servizio Kafka è stato già creato. Avendo creato il topic con due partizioni, ogni consumer avrà la ownership di una delle due partizioni.

Data Retrieval

Il servizio Data Retrieval si occupa di:

1. Connettersi al database del servizio db. Se la connessione fallisce ritenta ogni 5 secondi.
2. Esporre una interfaccia REST, accessibile dall’utente, che restituisca i risultati delle seguenti query:
    - Nome di tutte le metriche monitorate dal servizio ETL Data Pipeline (/metriche);
    - Valori di stazionarietà, stagionalità e autocorrelazione per una data metrica (/<metrica>/metadati);
    - Valori di massimo, minimo, media e deviazione standard per le ultime 1, 3 e 12 ore per una data metrica (/<metrica>/statistiche);
    - Valori di massimo, minimo e media predetti per i prossimi 10 minuti per una data metrica considerando i campioni delle ultime 1, 3 e 12 ore (/<metrica>/predizioni).

Il servizio Data Retrieval è deployato in un container chiamato “data_retrieval”. Tale container, come specificato nel dockerfile, espone la porta 40000. Nel docker compose questa container port è mappata sulla host port 40000 affinché gli endpoint definiti possano essere raggiunti dall’utente. Il container viene avviato successivamente al container db poiché le sue funzionalità dipendono dal database.

SLA Manager

Il servizio SLA Manager si occupa di:

1. Connettersi al database del servizio db. Se la connessione fallisce ritenta ogni 5 secondi.
2. Esporre una interfaccia REST, accessibile dall’utente, che permetta di:
    - Settare l’SLA Set con i relativi SLO tramite una richiesta HTTP POST (/set) contenente un opportuno file json nel seguente formato: *{“metrica1”: {“max”: x, “min”: y}, “metrica2”: {“max”: w, “min”: v}, “metrica3”: {“max”: z, “min”: i}, “metrica4”: {“max”: h, “min”: j}, “metrica5”: {“max”: k, “min”: l}}*. È richiesto che vengano inserite esattamente 5 metriche e che per ognuna di esse venga specificato almeno uno tra il valore minimo e il valore massimo.
    - Ottenere lo stato dell’SLA Set, ovvero se per ogni metrica è stata registrata una violazione nelle ultime 1, 3 e 12 ore (/stato).
    - Ottenere il numero di violazioni per ogni metrica dell’SLA Set nelle ultime 1, 3 e 12 ore (/violazioni_passate).
    - Ottenere la previsione di possibili violazioni per ogni metrica dell’SLA Set considerando i campioni delle ultime 1, 3 e 12 ore (/violazioni_future).

Il servizio SLA Manager è deployato in un container chiamato “sla_manager”. Tale container, come specificato nel dockerfile, espone la porta 45000. Nel docker compose questa container port è mappata sulla host port 45000 affinché gli endpoint definiti possano essere raggiunti dall’utente. Il container viene avviato successivamente ai container db e etl_data_pipeline poiché le sue funzionalità dipendono da tali servizi.

Init Kafka

Il servizio Init Kafka si occupa di creare all’interno del servizio Kafka un topic “prometheusdata” con 2 partizioni e termina subito dopo. Il servizio è deployato in un container chiamato “init_kafka” che viene avviato successivamente al container kafka per ovvie ragioni.

Istruzioni per l'uso

Tutti i file necessari per l'avvio del progetto tramite Docker sono contenuti nella cartella corrente ("Progetto DSBD"). In particolare, tale cartella contiene:

- Il docker compose per l'avvio del progetto;
- Un file db.sql per configurare il database contenuto nel container db;
- Una cartella per ognuno dei servizi presentati contenente il dockerfile per la creazione dell'immagine del container e lo script python da eseguire.

Per avviare il progetto, dopo essersi posizionati nella cartella contenente il file docker-compose.yml, si deve lanciare il comando *docker compose up -d*.
Dopo l'avvio di tutti i container, da browser è possibile interagire con il sistema attraverso i seguenti URL:

- http://localhost:40000/metriche per contattare il servizio Data Retrieval e ottenere l'elenco di tutte le metriche monitorate;
- http://localhost:40000/<metrica>/metadati per contattare il servizio Data Retrieval e ottenere i metadati (stazionarietà, stagionalità, autocorrelazione) della metrica specificata;
- http://localhost:40000/<metrica>/statistiche per contattare il servizio Data Retrieval e ottenere le statistiche (max, min, avg, dev_std) della metrica specificata per le ultime 1, 3 e 12 ore;
- http://localhost:40000/<metrica>/predizioni per contattare il servizio Data Retrieval e ottenere massimo, minimo e media predetti per i successivi 10 minuti della metrica specificata considerando i campioni delle ultime 1, 3 e 12 ore;
- http://localhost:45000/set per contattare il servizio SLA Manager inviando una richiesta HTTP POST contenente un json con l'SLA Set e i relativi SLO da impostare. Il json inviato deve essere nel seguente formato: *{"metrica1": {"max": x, "min": y}, "metrica2": {"max": w, "min": v}, "metrica3": {"max": z, "min": i}, "metrica4": {"max": h, "min": j}, "metrica5": {"max": k, "min": l}}*. È richiesto che vengano inserite esattamente 5 metriche e che per ognuna di esse venga specificato almeno uno tra il valore minimo e il valore massimo;
- http://localhost:45000/stato per contattare il servizio SLA Manager e ottenere lo stato dell'SLA Set, ovvero vedere se per ogni metrica è stata registrata una violazione nelle ultime 1, 3 e 12 ore;
- http://localhost:45000/violazioni_passate per contattare il servizio SLA Manager e ottenere il numero di violazioni per ogni metrica dell'SLA Set nelle ultime 1, 3 e 12 ore;
- http://localhost:45000/violazioni_future per contattare il servizio SLA Manager e ottenere la previsione di possibili violazioni per ogni metrica dell'SLA Set considerando i campioni delle ultime 1, 3 e 12 ore.

Nella cartella /ETL_Data_Pipeline è presente il file log.txt, in cui sono memorizzati i tempi di esecuzione delle funzionalità del servizio ETL Data Pipeline.