Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di laurea in Informatica

Data ingestion e anomaly detection: il ciclo di vita del dato

Tutor Interno
Lamberto Ballan
Università di Padova

Tutor Esterno
Michele Giusto

Studente Candidato
Giovanni Sorice

Anno Accademico 2018 - 2019

Dedica.

Sommario

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduzione

In questo capitolo verrà introdotta l'azienda ospitante e le tecnologie da essa utilizzatel.

1.1 L'azieda

Data Reply S.r.l fa parte delle aziende del gruppo Reply S.p.a e si occupa del mondo Big Data, Data Science e Artificial Intelligence. L'azienda è giovane sia nella sua creazione, dato che è stata fondata nel 2013, sia dal punto di vista della composizione del personale. Questo le permette di avere flessibilità e freschezza mentale (in termini di idee) ma allo stesso tempo know how e conoscenza del settore messa in campo dagli elementi con maggiore esperienza.

Il mercato di Data Reply è quello della consulenza, un mondo spesso ostico e molto complesso ma che ha visto un'importante crescita degli investimenti in Italia negli ultimi anni.

Figura 1.1: Logo Data Reply S.r.l.

1.1.1 Le tecnologie da esplorare

Le principali tecnologie utilizzate dall'azienda riguardano i settori Big Data, Data Science e Artificial Intelligence. Troviamo un importante utilizzo delle suite di prodotti cloud, come AWS e Google Cloud, e di prodotti da poter utilizzare all'interno di macchine proprie, come Cloudera. Infine anche l'utilizzo di framework per il calcolo distribuito, come Apache Spark, sono di fondamentale importanza.

1.2 Processi del progetto

1.2.1 Metodologia di sviluppo

All'interno di Data Reply vengono utilizzati diverse metodologie di sviluppo, che vanno dalle metodologie agile, nella quale vediamo tra le più utilizzate Scrum e Kanban, a metodologie a cascata. Essendo un'azienda di consulenza, spesso si trova a lavorare anche con altre aziende allo stesso progetto e per questo motivo non è raro trovare metodologie più rigide ai cambiamenti come le metodologie a cascata.

1.2.2 Strumenti di supporto

Gli strumenti di supporto utilizzati, come le metodologie di sviluppo, sono decisi in base al progetto, al cliente e in accordo con le altre aziende che lavoreranno al progetto stesso. Di consueto, l'azienda propone l'utilizzo di Git come strumento di versionamento, GitLab per la gestione delle repository di lavoro e GitLab Mattermost per le comunicazioni informali tra i componenti del progetto e dell'azienda.

1.2.3 Propensione alla modernizzazione

L'azienda cerca sempre di rinnovarsi e rimanere al passo con le ultime tecnologie in modo da poter proporre ai propri clienti il giusto compromesso tra novità e affidabilità. Spesso infatti, vi sono progetti mirati allo studio ed utilizzo di tecnologie e prodotti da poco sul mercato, così da garantire un vantaggio competitivo sulle altre aziende. Questo si può anche notare dai raduni svolti dall'azienda madre Reply nella quale Data Reply cerca sempre di portare progetti innovativi e che vadano a toccare le ultime tendenze del momento.

2
Evoluzione dello stage

In questo capitolo verrà introdotto il progetto, spiegandone vincoli, obiettivi e motivi della scelta.

2.1 La proposta di stage

La proposta di stage mi è stata fatta nei mesi successivi alla visita all'azienda svolta attraverso l'Università di Padova. Fin da subito il complesso e lo spirito aziendale mi sono sembrati adatti ad ospitare degli studenti volenterosi di sperimentare nuove tecnologie.
Durante i colloqui con il Dott. Michele Giusto, mi è stato esposto un progetto incentrato sui Big Data e il rilevamento delle anomalie che corrispondeva al mio bisogno di affrontare nuove sfide e approfondire argomenti non trattati durante i corsi universitari.
Negli ultimi anni infatti, tra le tecnologie che hanno riscosso più successo e fama, troviamo i Big Data e tecniche di intelligenza artificiale con particolare attenzione al machine learning. Per questo motivo e per il mio già precedente interessamento verso queste tecnologie, ho deciso di dare forma agli obiettivi di questo progetto.

2.2 Il progetto

Il progetto si prefissava l'obiettivo di analizzare e costruire l'intero ciclo di vita del dato, secondo metodi, strumenti e tecniche innovative.

Per essere maggiormente concreti, il progetto prevedeva l'implementazione di un sistema di ingestion dei dati aziendali del cliente, che quindi comportasse l'estrapolazione di informazioni dal dato grezzo, con successiva classificazione degli utenti che interagivano con il sistema stesso, in modo da intercettare eventuali comportamenti sospetti.
Per svolgere questo progetto, prima del mio arrivo e dell'effettiva proposta, l'azienda ospitante ha svolto uno studio di fattibilità in modo da sottolineare eventuali carenze di informazioni, criticità e punti a favore del progetto.
Durante il mio periodo di permanenza, il progetto è stato portato al termine seguendo le indicazioni iniziali e con alcuni aggiustamenti durante il percorso. Inoltre, sono state rispettate le fondamentali indicazioni del tutor aziendale che mi ha affiancato durante lo stage.
Terminato il mio stage il prodotto risultava conforme alle aspettative e facilmente manutenibile grazie alla documentazione redatta durante il periodo di studio del problema, sviluppo e test.

2.2.1 Obiettivi

Durante i colloqui di esposizione del progetto di stage, sono sorti alcuni obiettivi da soddisfare al termine dello stesso.
Essi si dividono in:

- Tutti gli obiettivi che sono fondamentali per la corretta riuscita del progetto, anche detti, obiettivi obbligatori:
  - Il sistema deve essere in grado di aggiornare periodicamente la concezione di normalità, per adattarsi al contesto dinamico;
  - Il sistema deve essere in grado di ricevere i flussi batch ed archiviarli dove desiderato;
  - Il sistema deve essere in grado di processare i flussi batch producendo i dataset di output richiesti;
  - I processi devono essere programmati per avviarsi periodicamente.

- Obiettivi desiderabili, cioè, gli obiettivi che porterebbero ad una maggiore completezza del progetto di stage ma che non sono fondamentali per la sua buona riuscita:
  - Il sistema deve riuscire ad aggiornarsi continuamente senza interrompere l'erogazione del servizio;

  - Il sistema deve essere in grado di scrivere i risultati su una struttura interrogabile via SQL;
  - Il sistema deve poter scalare le risorse in base al volume di dati.

- Tutti gli obiettivi che sono marginali al progetto ma che porterebbero comunque valore aggiunto allo stesso, anche detti, obiettivi facoltativi:

  - Il sistema deve poter lavorare su dati archiviati su Elasticsearch;
  - Il sistema deve poter essere in grado di scrivere i propri output su DB NoSQL.

2.2.2 Vincoli

Durante i colloqui di esposizione del progetto di stage, sono sorti alcuni vincoli da soddisfare per svolgere al meglio lo stesso.

Vincoli temporali

La durata dello stage doveva essere al minimo di 300 ore e al massimo di 320, in modo da rispettare i crediti stabiliti all'interno del piano di studi, svolte tutte presso la sede in via Robert Koch 1/4 a Milano negli orari 9.00-13.00, 14.00-18.00.

Vincoli tecnologici

Utilizzo della suite Google Cloud per lo sviluppo dell'ingestion dei dati e di Spark MLlib per la parte di Machine Learning.

2.3 Motivi della scelta

La scelta di abbracciare il progetto di Data Reply è dovuto principalmente a 4 motivi:

- Proposta di un progetto interessante con esperti del settore;
- Azienda dinamica e giovane con grandi ambizioni;

- Argomenti e tecnologie molto ricercate nell'ultimo periodo che mi permetteranno di inserirmi al meglio nel mondo del lavoro;
- Possibilità di svolgere lo stage presso la sede di Milano, città molto dinamica e innovativa.

Questi motivi mi hanno convito a scegliere questa proposta per mettere alla prova le mie abilità.

3

Analisi e progettazione

In questo capitolo analizzeremo le metodologie di lavoro, i problemi riscontrati e i risultati ottenuti.

3.1 Metodo di lavoro

Dopo un'attenta analisi delle esigenze di questo progetto, la metodologia che abbiamo deciso di utilizzare è Agile Kanban. Questa metodologia, ci ha permesso di avere un'alta flessibilità sui task da svolgere, una visione ampia del progetto e il focus su una singola attività alla volta. La metodologia Kanban è stata scelta anche per affrontare l'esiguo numero del team, composto da sole 2 persone, con effettivamente solo una che lavorava a pieno regime su di esso. Questo perché rendeva un'ottima visione dell'insieme delle attività da svolgere, in progresso e già svolte, oltre a non necessitare di ruoli predefiniti come nell'Agile Scrum

Metodologia kanban Fa parte della famiglia delle metodologie agile, alla base vi è l'utilizzo di una lavagna nella quale vi è specificato uno stato per ogni colonna presente. Quello sopra riportato è uno degli esempi più semplici di lavagna Kanban, nella quale ogni task deve assumere tre stati per considerarsi completato "ToDo", "Doing" e "Done". Solitamente ogni componente del team ha la responsabilità di portare a completamente un solo task alla volta, così da evitare sovrapposizioni e confusione.

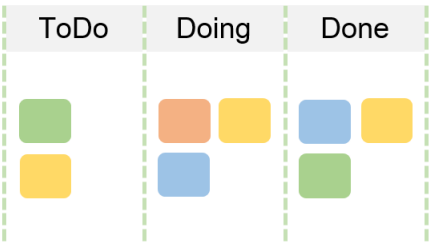

Figura 3.1: Esempio lavagna Kanban.

Questa metodologia ha sicuramente giovato al progetto dato l'alto livello di flessibilità garantito e necessario al rispetto dalle richieste periodiche del cliente.

3.2 Problemi affrontati

Affrontare sfide, problemi e quesiti di carattere tecnologico, architetturale e implementativo è stata sicuramente la parte più formativa del periodo di stage, dato che in queste situazioni lo studio autonomo e il confronto con esperti del settore come il mio Tutor sono stati di primaria importanza.
I problemi affrontati durante questo progetto sono molteplici e vanno dall'analisi e scelta delle tecnologie da utilizzare all'integrazione tra le stesse.
Per circoscrivere le problematiche abbiamo deciso di sviluppare il progetto in due sotto-progetti autonomi. Il primo tratta dell'ingestion dei dati e il secondo dell'utilizzo dei dati con il fine di categorizzare gli utenti. Da questi quindi, derivano diverse tipologie di problemi, tra cui la scelta e lo studio degli strumenti e delle best practices.
Il punto cruciale del sotto-progetto riguardante l'ingestion è stato lo studio delle tecnologie riguardanti Google Cloud Platform e con esso le conseguenti scelte obbligate, come lo studio di Apache Beam, l'utilizzo di sole librerie compatibili con Google Cloud ed i suoi strumenti.
Per quanto riguarda la classificazione degli utenti, si è scelto di intraprendere una strada che potesse portare a dei vantaggi di utilizzo e portabilità, cioè si è deciso di utilizzare un container Docker per lo sviluppo ed utilizzo degli algoritmi di classificazione. Questo è stato fatto per agevolare l'installazione e configurazione in locale dell'intero pacchetto di strumenti da utilizzare, tra cui Spark, Kafka e Hadoop. Ovviamente, l'utilizzo di Docker ha comportato la scelta del miglior compromesso per la creazione di un container adatto. Dato l'ampio pacchetto di strumenti da studiare in autonomia, abbiamo deciso quindi di utilizzare un container Cloudera già contenete la maggior parte degli strumenti necessari per sviluppare

un processo di classificazione degli utenti. Inoltre, per far si che i dati venissero continuamente aggiornati, come effettivamente succede nella fase di ingestion, abbiamo deciso di utilizzare Kafka Straming per mantenere un flusso continuo di verifica delle anomalie, così da mantenere in costante aggiornamento il cliente.

3.3 Requisiti

3.3.1 Struttura

Ogni requisito è descritto dalla seguente struttura:

- Tipo;
- Importanza;
- Stato implementazione;
- Fonti.

Inoltre, a ciascun requisito corrisponde un codice identificativo cosi composto:

R {importanza}.{tipo}.{identificativo}

- R specifica che si tratta di un requisito;
- importanza identifica la rilevanza del requisito e può assumere 3 valori:
  - 0: indica che il requisito è obbligatorio e il suo soddisfacimento dovrà necessariamente avvenire;
  - 1: indica che il requisito è desiderabile, cioè il suo soddisfacimento può portare maggiore completezza al sistema ma non è fondamentale per lo stesso;
  - 2: indica che il requisito è opzionale, e quindi la decisione di implementarlo o meno verrà presa dopo le dovute considerazioni.

- tipo distingue se si tratta di un requisito funzionale (F), di qualità (Q), di prestazione (P) o di vincolo (V);
- identificativo è un numero progressivo che identifica i sottocasi.

3.3.2 Tabella dei requisiti

Per assolvere gli obiettivi del progetto e per rispondere in modo concreto ai problemi sorti, durante lo stage sono stati definiti i seguenti requisiti.

Id Requisito	Descrizione	Fonte
R0F1	Il sistema deve essere in grado di trasferire i file con estensione evtx e xml dal server SFTP a Google Cloud Storage.	Interno
R1F1.1	Il sistema deve essere in grado di verificare la corretta estensione dei file.	Interno
R1F1.2	Il sistema deve essere in grado di scartare i file con una non corretta estensione.	Interno
R1F1.3	Il sistema deve essere in grado di validare i file con estensione evtx.	Interno
R1F1.4	Il sistema deve essere in grado di convertire i file evtx in xml.	Interno
R0F2	Il sistema deve essere in gardo di aggiornarsi periodicamente.	Interno
R1F2.1	Il sistema deve essere in gardo verificare la presenza di nuovi file all'interno del server SFTP.	Interno
R1F2.2	Il sistema deve essere in gardo di avviare il processo di ingestion dei file ad ogni nuovo arrivo.	Interno
R1F2.2.1	Il sistema deve essere in gardo di notificare l'arrivo di nuovi file all'interno del Bucket Google Cloud Storage.	Interno
R1F2.2.2	Il sistema deve essere in gardo di avviare il processo di ingestion una volta ricevuta la notifica dal Bucket Google Cloud Storage.	Interno

Id Requisito	Descrizione	Fonte
R0F3	Il sistema deve essere in grado di processare i flussi batch producendo i dataset di output richiesti.	Interno
R0F3.1	Il sistema deve essere in gardo di archiviare i file di output all'interno del Bucket Google Cloud Storage.	Interno
R1F3.2	Il sistema deve essere in gardo di archiviare i file di output all'interno di Google Cloud BigQuery.	Interno
R0F4	I processi devono essere schedulati per avviarsi periodicamente.	Interno
R1F5	Il sistema deve riuscire ad aggiornarsi continuamente senza interrompere l'erogazione del servizio.	Interno
R1F6	Il sistema deve essere in grado di scrivere i risultati su una struttura interrogabile via SQL.	Interno
R1F7	Il sistema deve essere in grado di classificare i dati all'interno dei file provenienti dal processo di ingestion.	Interno
R1F7.1	Il sistema deve essere in grado di classificare i dati all'interno dei file provenienti dal processo di ingestion per singolo giorno.	Interno
R1F7.2	Il sistema deve essere in grado di classificare i dati all'interno dei file provenienti dal processo di ingestion per singolo file su base multi-giorno.	Interno
R1F8	Il sistema deve essere in grado di inviare una email al manager responsabile in caso di anomalie nei dati classificati.	Interno
R1F9	Il sistema deve essere in grado di rendere visibile dei report all'interno di Google Data Studio.	Interno

Id Requisito	Descrizione	Fonte
R2F10	Il sistema deve poter lavorare su dati archiviati su Elasticsearch.	Interno
R2F11	Il sistema deve poter essere in grado di scrivere i propri output su DB NoSQL.	Interno
R0Q1	La progettazione e il codice devono seguire le norme riportate all'interno degli standard di Data Reply S.r.l.	Interno
R0Q2	Tutti i documenti e il codice prodotto devono rispettare le metriche riportate all'interno degli standard di Data Reply S.r.l.	Interno
R0Q3	Deve essere prodotto un manuale sviluppatore.	Interno
R0V1	Il sistema deve utilizzare strumenti appartenenti a Google Cloud Platform o ad esso compatibili.	Interno
R1V1	Il sistema deve utilizzare Spark MLlib per il sistema di classificazione.	Interno
R2V1	Il sistema di classificazione deve essere sviluppato all'interno di un container Docker.	Interno
R1P1	Il sistema deve poter scalare le risorse in base al volume di dati.	Interno

Tabella 3.1: Requisiti di Vincolo

3.4 Pianificazione del lavoro

Durante i colloqui svolti con il tutor aziendale, è stato redatto il piano di lavoro. Ciò ha portato la suddivisione dello stage in 8 parti, ognuna della durata di una settimana.

Prima Settimana (40 ore)

- Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;
- Verifica credenziali e strumenti di lavoro assegnati;
- Presa visione dell'infrastruttura esistente;
- Formazione sulle tecnologie adottate.

Seconda Settimana (40 ore)

- Comprensione strumenti di storicizzazione;
- Analisi funzionamento filesystem distribuito;
- Produzione script Python per produzione dati simulati.

Terza Settimana (40 ore)

- Caricamento dati su filesystem distribuito;
- Securizzazione accesso ai dati;
- Studio teorico approccio statistico al problema.

Quarta Settimana (40 ore)

- Comprensione strumenti di processamento;
- Analisi funzionamento RDD;
- Implementazione processo batch con Python e Spark.

Quinta Settimana (40 ore)

- Analisi funzionamento Dataframe;

- Implementazione processo di elaborazione Spark;
- Ingegnerizzazione soluzione batch.

Sesta Settimana (40 ore)

- Test prestazionale processo spark;
- Tuning prestazionale;
- Studio applicazione real-time.

Settima Settimana (40 ore)

- Storicizzazione dati su storage persistente;
- Studio funzionamento Kafka;
- Implementazione processo real-time con Spark Streaming.

Ottava Settimana - Conclusione (40 ore)

- Costruzione layer accesso al dato persistente;
- Ingegnerizzazione soluzione Spark Streaming;
- Integrazione storage persistente.

possibili diagrammi

3.5 Risultati

Alla fine del periodo di stage, i risultati ottenuti sono stati considerati più che positivi dal tutor aziendale Dott. Michele Giusto, dato che sono stati soddisfatti tutti i requisiti obbligatori e desiderabili. I requisiti facoltativi, non sono stati implementati per lasciare maggiore spazio e importanza ai requisiti obbligatori.
Inoltre, il sotto-progetto riguardante la fase di ingestion ha avuto ottimi risultati, tale per cui si è dimostrata pronta per essere utilizzata in fasi di ”produzione”. Per quanto riguarda la parte di classificazione, è stata messa a punto per scheletro e funzionalità senza però essere pronta per gestire effettivamente l'aggiornamento real time dei dati, soprattutto perché da parte del clienti vi è stato un rallentamento nel passaggio di dati ed informazioni per la scelta del miglior modello.

4
Sviluppo

In questa sezione, verranno elencate e introdotte le tecnologie che sono state utilizzate nel progetto di stage.

4.1 Tecnologie

4.1.1 Google Cloud

Il principale servizio utilizzato durante il progetto è stato Google Cloud, in esso troviamo una moltitudine di strumenti utili allo sviluppo, all'esecuzione e all'interazione di progetti riguardanti Big Data e Intelligenza Artificiale. Inoltre, importante risulta la possibilità di consultare, manipolare e presentare i dati. Il progetto Google Cloud (anche detto Google Cloude Platform), viene lanciato nel 2008 con uno dei suoi servizi di punta ancora oggi App Engine. Nell'arco degli anni sia il bacino di utenti, sia i servizi offerti sono via via aumentati. Oggigiorno i suoi clienti sono molti, tra cui grandi aziende, e le funzionalità di spicco sono molteplici, i campi di maggiore interesse riguardano strumenti di Compute, Storage & Databases, Big Data, Cloud AI e IoT. Per quanto riguarda l'esperienza personale, ho trovato Google Cloud un buon servizio, con dei problemi da risolvere sulla completezza della documentazioni e a volte con grafiche da migliorare, ma comunque un ottimo compromesso tra qualità e prezzo. Il resto della sezione introduce il prodotti e i servizi GCP utilizzati durante lo stage.

Figura 4.1: Logo Google Cloud Platform.

Google Cloud Storage

Lanciato nel 2010, Google Cloud Storage è il servizio di hosting dei file della piattaforma. All'interno ogni utente può scegliere in che modalità salvare i dati (zona del mondo, carico di lavoro e frequenza d'accesso), così da trovare la soluzione maggiormente adatta ai propri bisogni. I suoi punti di forza sono:

- Interoperabilità: possibilità di interfacciarsi con strumenti e servizi di aziende terze (come Amazon S3);
- Consistenza: tutte le operazioni sono svolte secondo il principio di atomicità;
- Controllo di accesso: possibilità di definire l'accesso ad un singolo oggetto o bucket a seconda della categoria di appartenenza degli utenti;
- Resumable Uploads: in caso di fallimento di un upload, questo può essere continuato/ricominciato.

Per quanto riguarda l'esperienza d'utilizzo, si è rivelato un ottimo servizio di storage, l'abilitazione di un'unica API rende disponibile l'intero pacchetto di funzionalità e ciò l'ho trovato molto comodo.

Google Cloud BigQuery

Servizio lanciato insieme a Google Cloud Storage nel 2010, Google Cloud BigQuery, è lo strumento che permette di interagire e analizzare grandi moli da dati, definito come un data

warehouse. All'interno di GCP, vi è un'apposita interfaccia utente per utilizzarlo. Si basa su tabelle codificate in formato JSON e sulla quale vengono processate query in standard SQL tramite avanzati algortmi di MapReduce.
I principali punti di forza sono:

- La piattaforma rende disponibile la creazione e distruzione di tabelle basati su schemi JSON, inoltre rende disponibile l'import e l'export di dati in formati come CSV e JSON;
- La semplicità del linguaggio SQL con scalabilità e prestazioni che solo i servizi cloud riescono a dare;
- Un'ottima integrazione con servizi e API esterni, per la raccolta e l'utilizzo dei dati;
- Possibilità di garantire l'accesso ad uno o più categorie di utenti;
- Possibilità di interagire con strumenti di Machine Learning nativi di Google e non (come TensorFlow).

L'opinione del team di sviluppo riguardo lo strumento è molto positiva sopratutto grazie alla scalabilità automatica che rende semplice e veloce l'utilizzo delle query su grandi moli di dati, un possibile punto di debolezza può essere trovato nell'utilizzo in sola lettura dei dati esistenti, quindi non vi è la possibilità di modifica dei record se non un nuovo inserimento.

Google Cloud Dataflow

Rilasciato al pubblico nel 2015, Google Cloud Dataflow è lo strumento che permette di processare pipeline per l'integrazione, la preparazione e l'analisi di grandi quantità di dati. Il modello di pipeline consigliato è quello di Apache Beam che vedremo in seguito.
Tra i maggiori punti di forza troviamo:

- Creazione di pipeline secondo modelli già esistenti o personalizzati dall'utente;
- Composizione di pipeline, e quindi job specifici, direttamente da codice;
- Possibilità di utilizzo di dati da molteplici fonti, non solo da strumenti Google;
- Ottima scalabilità interna in base agli effettivi bisogni del lavoro in esecuzione;
- Buona granularità della specificazione dei parametri di esecuzione della pipeline;
- Visualizzazione grafica della pipeline (Possibile immagine?????).

Per quanto concerne l'utilizzo di questo prodotto, il team si esprime con un giudizio più che positivo dato anche la grande quantità di documentazione disponibile.

Google Data Studio

Lanciato nel 2016, Google Data Studio è lo strumento di creazione e visualizzazione dei dati attraverso report e dashboard. L'idea che sta alla base di questo tool è la condivisione dei dati che sono presenti all'interno di GCP attraverso tabelle e grafici.
I maggiori vantaggi nell'utilizzarlo sono:

- Accesso ai dati personali di GCP;
- Possibilità di integrare i dati fra di loro con query standard SQL;
- Condivisione a gruppi di utenti.

Durante l'esperienza lo strumento si è rivelato molto acerbo nelle sue funzionalità e nell'utilizzo, oltre a dover attendere troppo tempo per il caricamento di dati e grafici. L'idea alla base è ottima, dato che vi è la possibilità di utilizzare i dati all'interno di GCP anche in continuo aggiornamento, d'altro canto la documentazione e l'interfaccia utente non rendono semplice quanto potrebbero la creazione di un report aziendale da presentare al cliente.

Google Cloud Function

Rilasciato in beta nel 2017, Google Cloud Functions è il perfetto intermediario tra gli eventi che vengono scaturiti nel cloud e l'invocazione di servizi API. Le origini degli eventi supportati sono:

- Cloud Pub/Sub;
- Cloud Storage;
- HTTP;
- Stackdriver Logging;
- Firebase.

Inoltre, per ogni tipologia di evento vi sono ulteriori specificazioni che non verranno trattate tutte in modo estensivo in questo elaborato.
Per quanto concerne l'utilizzo svolto durante il progetto, il team ritiene questo strumento valido e con ottime potenzialità e margini di miglioramento. Uno svantaggio molto importante da sottolineare è la scarsa documentazioni presente nelle fonti ufficiali.

Google Cloud Composer

https://cloud.google.com/composer/?hl=it

4.1.2 Apache NiFi

> An easy to use, powerful, and reliable system to process and distribute data.
>
> NiFi Creator

La descrizione che viene data nella documentazione di NiFi riassume alla perferzione le caratteristiche di questo strumento e ciò che ci permette di fare.
Innanzitutto, le sue funzioni sono proprio quelle di automatizzare lo "scorrere" del flusso dati all'interno del sistema in cui viene utilizzato. Le funzionalità sono molteplici e ben rodate, vanno dal semplice spostamento dei file, al filtraggio, composizione e conversione in diversi formati. La documentazione è molto esauriente, spiegando anche più del necessario tutti i passaggi da seguire per utilizzare un determinato componente. La semplicità e l'intuitività nel costruire il flusso di dati, rendono l'esperienza utente decisamente piacevole, anche grazie all'interfaccia grafica semplice e intuitiva. Altro punto a favore è l'affidabilità del prodotto e la grande quantità di componenti disponibili orientati alla sicurezza, così da rendere il passaggio di informazioni in più sicuro possibile.

Figura 4.2: Logo Google Cloud Platform

4.1.3 Apache Beam

Apache Beam è un implementazione del modello Dataflow, esso infatti fissa come definire ed eseguire pipeline sui dati di tipo ETL, batch e streaming continui. Questo progetto nasce dopo il rilascio di un SDK contenente un implementazione del modello Dataflow da parte di Google nel 2014, nel 2016 Google decise di concedere il core del proprio SDK per costituire il progetto OpenSource Apache Beam.

I concetti alla base di questa implementazione, e quindi del modello Dataflow, possono essere riassunti nel cercare un modello il quale riesca a definire ed eseguire pipeline di tipo batch e streaming allo stesso tempo, senza dover differenziare le pipeline come nel modello Lambda architecture.
Un tipico utilizzo di una pipeline può essere il seguente:

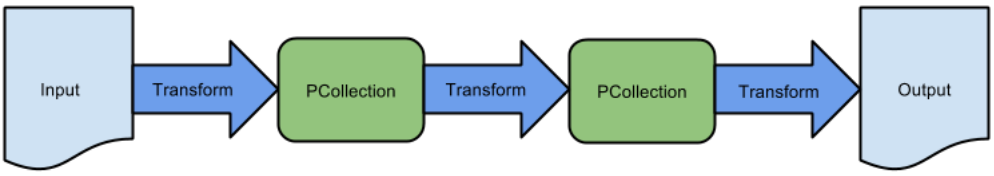

Figura 4.3: Esempio utilizzo Apache Beam.

- Creazione della Pipeline;
- Creazione di una PCollection da dei dati di input;
- Applicazione di PTrasforms sulla PCollection;
- Scrittura dei dati all'interno di un file.

Le azioni di maggiore interesse stanno nelle PTrasforms, infatti rende possibile processare ogni singolo elemento delle PCollection, effettuare raggruppamenti per uno o più elementi, unire e partizionare le PCollections. La documetazione a disposizione è decisamente idonea e precisa rispetto alla potenzialità di utilizzo. Inoltre, essendo un progetto OpenSource vi è la possibilità di collaborare al continuo sviluppo, e quindi al successo, che questa tecnologia sta avendo.

4.1.4 Apache AirFlow

Apache AirFlow è una piattaforma per creare, schedulare e monitorare data warehouse. Airflow, inoltre, cerca la soluzione più performante per l'esecuzione dei task, creando un Grafo Aciclico Diretto (meglio conosciuto come "data warehouse" cioe " Directed Acyclic Graphs") in modo da riconoscere le dipendenze tra i compiti stessi ed eseguire i task con meno dipendenze. La scelta di utilizzare questa tecnologia è stata presa sia per la già precedente esperienza aziendale con essa, sia perché completamente compatibile e già installata all'interno di

Google Cloud Composer. Data la sua notevole compatibilità ci ha permesso di soddisfare un requisito del cliente riguardante l'ingestion in tempo reale dei dati inviati.

4.1.5 Docker

4.1.6 Apache Spark

Apache Spark MLLib

4.2 Integrazione

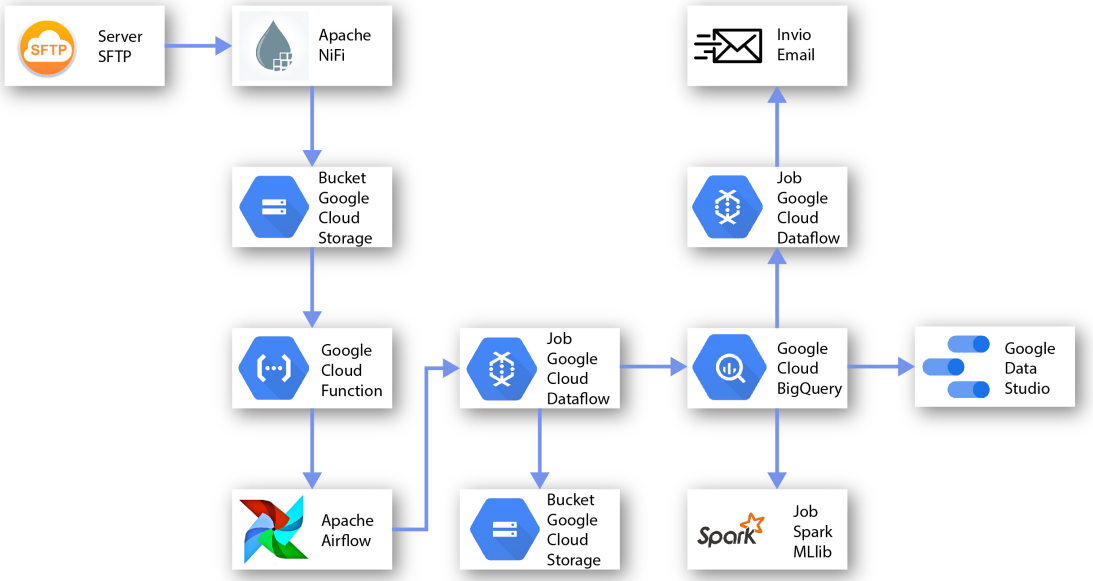

Figura 4.4: Workflow progetto.

L'integrazione delle tecnologie elencate alla sezione 4.1, ha portato a stabilire un vero e proprio ciclo di vita del dato. Infatti, partendo dai file che il cliente rilascia giornalmente all'interno di un server SFTP(1), tramite Apache NiFI (2), avveniva una prima pre elaborazione in base alla tipologia di file così da unificare il formato su cui lavorare.
Questi file vengono spostati all'interno di Google Cloud Storage (3), il quale, al trasferimento di ogni file, scatena l'avvio delle Google Cloud Function (4) che di conseguenza avviano il processo di ingestion sul file appena trasferito tramite l'attivazione del file DAGs all'interno di AirFlow (5).
L'ingestion è composta da alcuni job Google Cloud Dataflow (6) scritti in Python e carat-

terizzati dall'utilizzo di Apache Beam (7) per la creazione delle pipeline dei dati. Questi job, rendono i dati leggibili e consultabili. Dopodiché, vengono riportati all'interno di apposite tabelle Google Cloud BigQuery (8).
Al termine di questo processo, abbiamo il definitivo passaggio da semplice dato ad effettiva informazione per cliente tramite i report e le dashborad riassuntive di Google Data Studio (9).
Per concludere il processo da dato ad informazione, abbiamo costruito un sistema che, tramite Kafka (10) e quindi secondo lo schema produttore-consumatore in streaming, invia dei dati ad un algoritmo di Machine Learning sviluppato tramite Apache Spark (11)(più in particolare tramite la libreria Apache Spark MLlib) che classifica gli utenti su base settimanale, conserva le classificazioni e verifica che ogni utente non abbia cambi di classe in modo anomalo.

Come?

Durante l'esperienza lo strumento si è rilevato...

5 Conclusione

Analisi retrospettiva, cosa cambierei

Bibliografia

Acknowledgments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Glossario

Agile Scrum . 7

Apache Spark . 2

App Engine . 15

atomicità . 16

AWS a. 2

best practices . 8

bucket . 16

Cloudera . 2

container . 8

data warehouse . 20

data warehouse . 16

data warehouse . 20

ETL . 19

Git . 2

GitLab . 2

GitLab Mattermost . 2

Google Cloud a. 2, 5

ingestion . 4