Lake Formation: Data Security e Data Governance mediante LF-TBAC

9 Novembre 2021 - 10 min. read

AWS Glue | *AWS Identity and Access Management (IAM)* | *AWS Lake Formation* | *Data Security and Governance* | *Database*

Introduzione

Il Big Data è cresciuto rapidamente come modo per descrivere le informazioni ottenute da fonti eterogenee quando diventa incredibilmente difficile gestirle in termini di **varietà**, **veridicità**, **valore**, **volume** e **velocità**. Tuttavia, esso può essere considerato il "nuovo petrolio a causa del potenziale di generare valore aziendale".

Senza adeguate procedure di governance o di gestione della qualità, i data lake possono trasformarsi rapidamente in “paludi” di dati ingestibili. I data engineer sanno perfettamente che i dati di cui hanno bisogno vivono in questi data lake, ma senza una chiara strategia di governance dei dati, non saranno in grado di trovarli, fidarsi o utilizzarli in modo adeguato ai loro bisogni.

Una sfida molto comune è **mantenere** la governance, il **controllo degli accessi** sugli utenti che operano sul Data Lake e la protezione delle informazioni sensibili.

Le aziende devono centralizzare la governance, il controllo degli accessi e applicare una strategia supportata da servizi managed per controllare in modo granulare l'accesso degli utenti ai dati.

Affrontare queste problematiche richiede tipicamente due approcci: *manuale*, **più flessibile** ma **complesso**; *gestito* che richiede che la propria **soluzione si adatti a**

standard specifici, ma in cambio **elimina tutte le complessità di gestione** per gli sviluppatori.

Questo articolo guiderà il lettore attraverso la configurazione del proprio Data Lake mediante Lake Formation, mostrando tutte le sfide che devono essere affrontate durante il processo mostrando un occhio particolare alla sicurezza e alla governance attraverso l'approccio LF-TBAC.

Il controllo degli accessi basato su tag, in breve **TBAC**, è un modo sempre più diffuso per risolvere queste sfide, applicando vincoli basati su tag associati a risorse specifiche.

Quindi, senza ulteriori indugi, andiamo a dare un'occhiata!

Che cos'è l'accesso tramite TBAC

Il controllo degli accessi basato su tag consente agli amministratori di risorse abilitate per IAM di creare policy di accesso, basate su tag esistenti, associati a risorse idonee.

I provider cloud gestiscono i permessi sia degli utenti che delle applicazioni tramite delle policy, documenti con regole che fanno riferimento a risorse. Applicando i tag a tali risorse, è possibile definire condizioni di allow/deny semplici ed efficaci.

L'utilizzo dei tag di gestione degli accessi può ridurre il numero di criteri di accesso necessari all'interno di un account cloud, fornendo anche un modo semplificato per concedere l'accesso a un gruppo eterogeneo di risorse.

Perchè S3 da solo non basta

S3, come la maggior parte dei servizi AWS, **sfrutta i *principal* di IAM per la gestione degli accessi**, il che significa che è possibile definire quali parti di un bucket (file e cartelle o prefissi) un singolo principal IAM può leggere o scrivere; tuttavia non è possibile limitare ulteriormente l'accesso IAM a parti specifiche di un oggetto, né a determinati segmenti di dati archiviati all'interno di oggetti.

Ad esempio, supponiamo che i dati della nostra applicazione siano archiviati come una raccolta di file parquet suddivisi per paese in cartelle diverse.

È **possibile vincolare un utente ad accedere solo agli utenti appartenenti a un determinato paese**. Tuttavia, non c'è modo di impedire loro di leggere le informazioni anagrafiche (es. nome utente e indirizzo) memorizzate come colonne nel file parquet.

L'unico modo per impedire agli utenti di accedere a informazioni sensibili sarebbe crittografare le colonne prima di scrivere i file su S3, il che può essere **lento**, **ingombrante** e aprire un "vaso di Pandora" per quanto riguarda **l'archiviazione delle chiavi**, la condivisione e, infine, la **disattivazione delle chiavi**.

Inoltre, **concedere l'accesso a entità esterne utilizzando i principal IAM è spesso di per sé un problema non banale**.

Fortunatamente, AWS **offre una soluzione tutto-in-uno al problema delle autorizzazioni di Data Lake su S3**: facciamo entrare in gioco AWS Lake Formation!

AWS Lake Formation è un servizio completamente gestito che semplifica la creazione, la protezione e la gestione dei data lake, automatizzando molti dei complessi passaggi manuali necessari per crearli.

Lake Formation fornisce anche **il proprio modello di autorizzazioni, che è ciò che vogliamo esplorare in dettaglio, che potenzia il classico modello di autorizzazioni AWS IAM**.

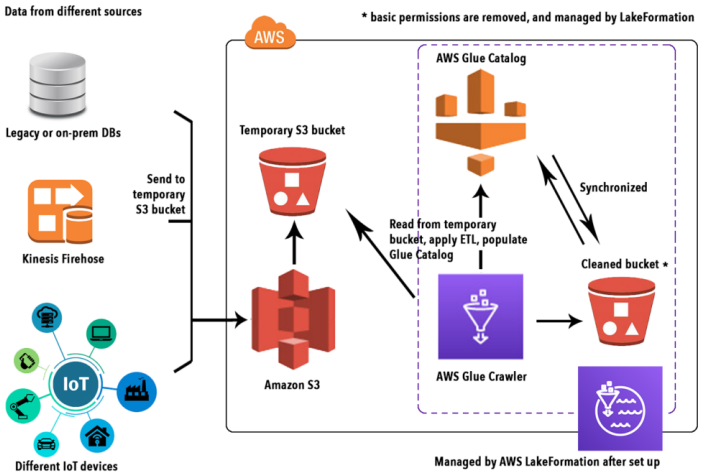
Questo modello di autorizzazioni definito in modo centralizzato, consente l'accesso granulare ai dati archiviati nei data lake attraverso un semplice meccanismo di concessione/revoca.

Quindi, sfruttando la potenza di Lake Formation, vorremmo dimostrare, con una soluzione semplice, come affrontare le suddette sfide di S3; proseguiamo!

Utilizziamo l’approccio TBAC in LakeFormation

Per accompagnare il lettore nella comprensione del motivo per cui AWS Lake Formation può essere una buona scelta nell'affrontare le complessità della gestione di un DataLake, abbiamo preparato un semplice tutorial su come migrare dati eterogenei.

Dai database locali legacy a S3, creando anche un catalogo Lake Formation per gestire la pulizia dei dati, le autorizzazioni e ulteriori operazioni.

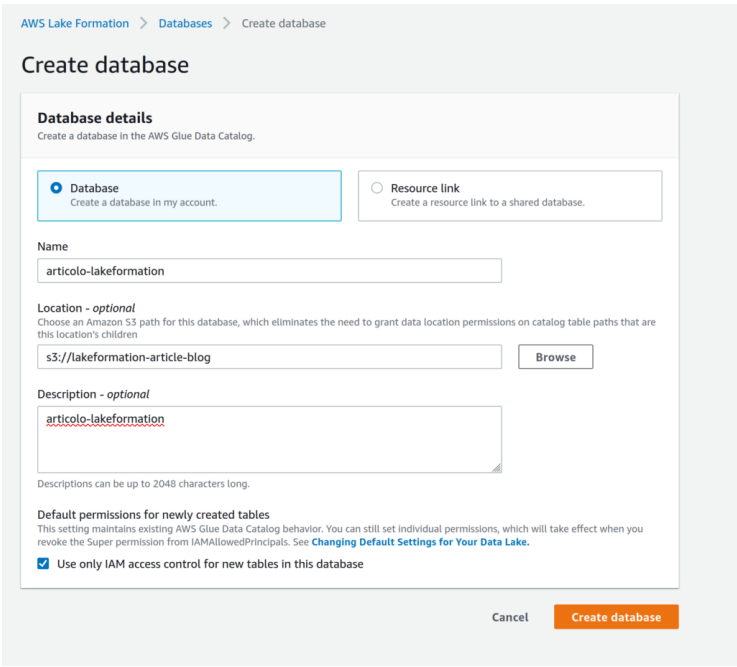


Schema di esempio del nostro tutorial

Migrazione di dati on-prem via AWS Glue

Il primo passo per la creazione di un Data Lake è ovviamente quello di recuperare, trasformare e inserire i dati. In questo semplice esempio, abbiamo utilizzato un set di dati utente simulato da un database MySQL. La colla AWS è il modo naturale per connettersi all'origine dati eterogenea, dedurre il loro schema importare e trasformare i dati e infine scriverli su S3 come spiegato in dettaglio qui.

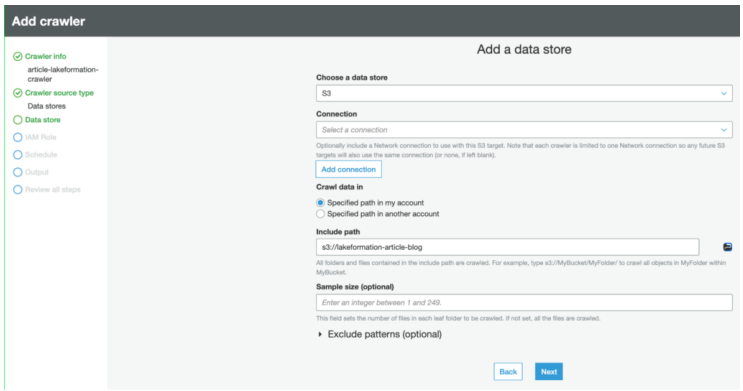
Dopo che i dati sono stati caricati in un bucket S3 temporaneo, è necessario creare **un database in Lake Formation** per connettersi a un **Glue Crawler** ed eseguirlo sul prefisso S3 per popolare un Glue Catalog per i dati.
Vai alla **console AWS Lake Formation**, nella pagina *Database* nella **schedaData catalog**, e inserisci un nome per il database e il tuo percorso su S3.



Creare un nuovo Database da Lake Formation

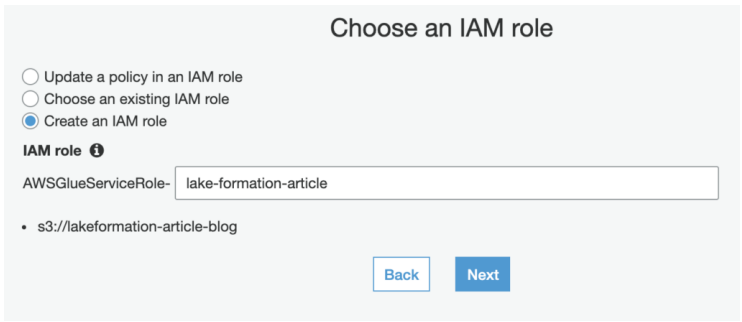
Nota: la creazione di un database direttamente da Lake Formation assicura che le autorizzazioni siano correttamente associate ad esso; avremmo potuto fare la stessa cosa anche da AWS Glue, ma avremmo avuto bisogno di uno sforzo aggiuntivo per modificare le autorizzazioni per i passaggi successivi.

Dopo che il database è stato creato, abbiamo bisogno di creare un Glue Catalog, che è un metastore contenente lo schema (schema-on-read) dei dati salvati su S3 (di solito come file parquet).
Avere uno schema Glue è **necessario per configurare i livelli di accesso da AWS Lake Formation per il proprio Data Lake su S3**. Per farlo, basta creare un crawler e collegarlo allo stesso percorso S3 del database e impostare **quel DB come output del crawler**.



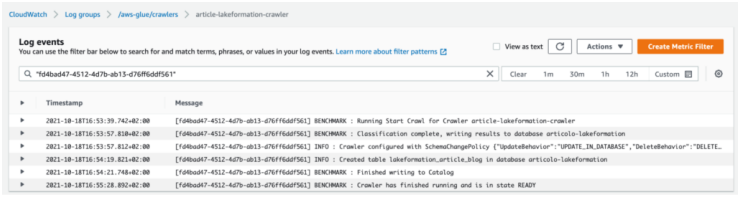
Setup di un AWS Glue Crawler standard

Per utilizzare il crawler, è necessario un ruolo IAM, ma fortunatamente AWS ha un step per questo nella procedura guidata di creazione del crawler:



Come creare uno IAM role da utilizzare per il Crawler

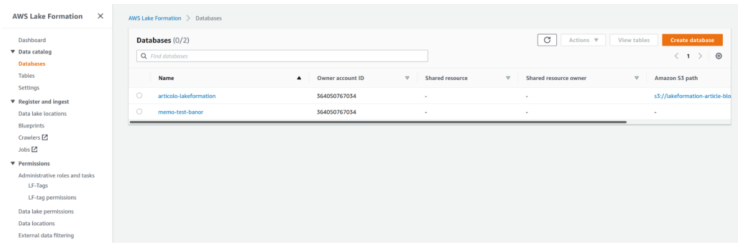
Non appena il Crawler è creato, e i dati importati nel catalogo, siamo pronti per il prossimo step.



Cloudwatch Logs dimostra che il Crawler ha funzionato correttamente

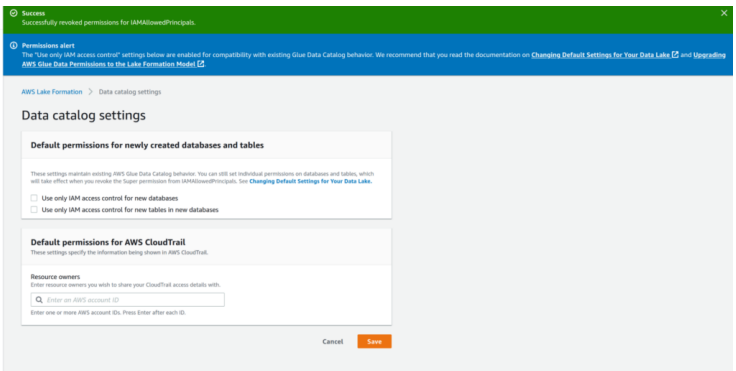
AWS Lake Formation

Avendo a disposizione un catalogo di dati di Glue, è il momento di configurare Lake Formation per gestire finalmente le autorizzazioni di accesso degli utenti.
Per fare ciò, iniziamo andando nella **dashboard di Lake Formation e rimuovendo i consueti permessi di accesso su S3**.



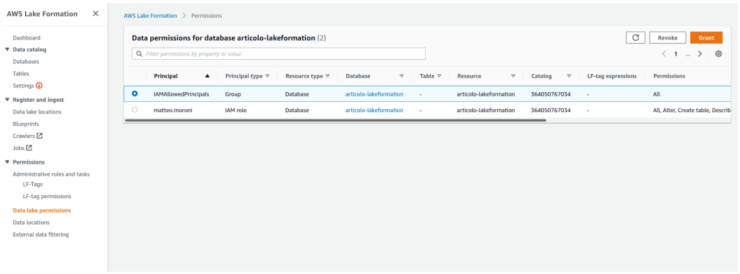
Dashboard di Lake Formation

Quindi possiamo andare su *Data Catalog Settings* e deselezionare *Use only IAM access control for new databases* e *Use only IAM access control for new tables in new databases.*
Per impostazione predefinita, l'accesso alle risorse del catalogo dati e ai percorsi su Amazon S3 è controllato esclusivamente dalle policy di AWS Identity and Access Management (IAM), deselezionando i valori rendiamo effettive le autorizzazioni individuali di Lake Formation.



Settaggi di Lake Formation data catalog: dobbiamo disabilitare entrambi i flag Use only

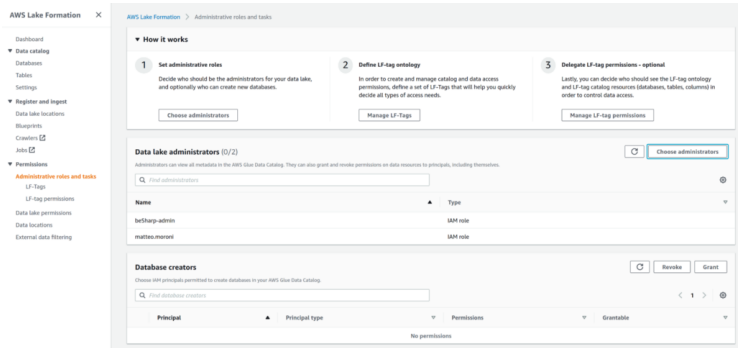
Una volta che le r**esponsabilità di accesso sono state delegate a Lake Formation**, possiamo rimuovere l'accesso per il gruppo IAM standard *IAMAllowedPrincipals*, nella scheda *Permissions* del data lake; selezioniamo dunque **permission of the IAM group** e facciamo click su *Revoke*.



Rimuoviamo le autorizzazioni standard IAMAllowedPrincipals

L'utente che crea il data lake sarà anche elencato in questa sezione con i privilegi di amministratore; se si desidera invece che l'utente mantenga l'accesso ai dati è possibile lasciare l'autorizzazione così com'è, altrimenti è possibile **revocare o limitare l'autorizzazione all'utente/ruolo.**

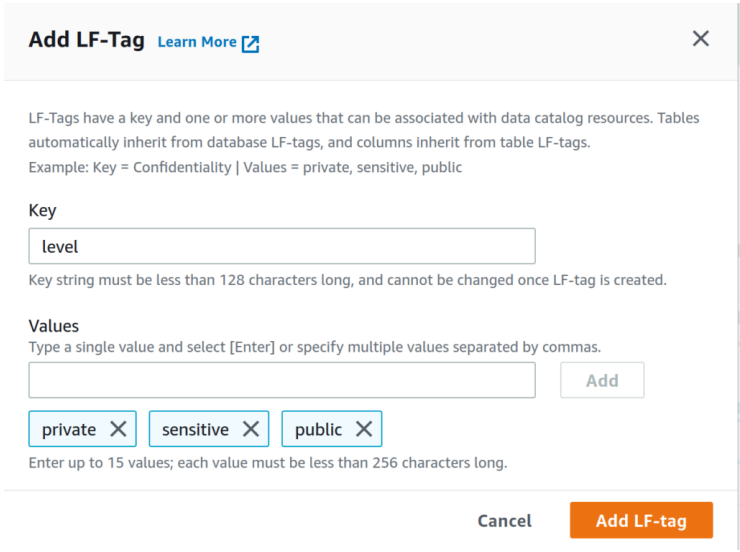
Nota: se è necessario aggiungere un Data lake administrator principal, è possibile farlo accedendo ad Administrative roles and tasks e aggiungendo un ***amministratore del Data Lake****.*



Aggiungiamo admin e db creator dalla console

Una volta completati tutti questi passaggi, è il momento di iniziare a definire i tag Lake Formation (da ora in poi **LF-Tags**), che verranno utilizzati per limitare l'accesso al data lake.

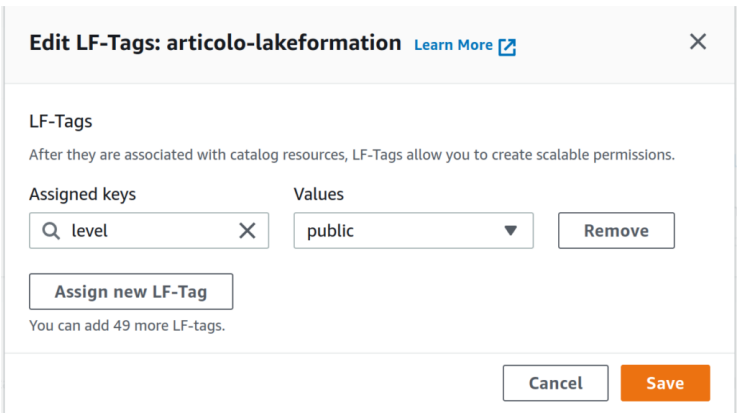
Dalla pagina *LF-Tag* sotto la scheda *Permissions* **creiamo un nuovo LF-Tag**, come chiave utilizziamo *level*, come valori invece *private*, *sensitive* e *public* separati da virgola proprio come in figura. Facciamo infine click su **Add LF-tag**.



Creazione di un LF-Tag

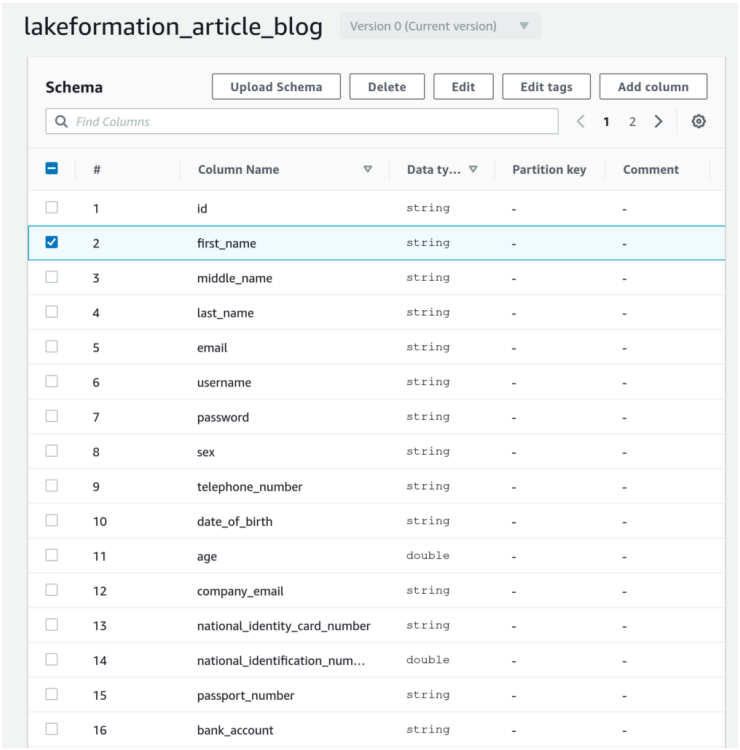
Ora, una volta creati, come possiamo utilizzare questi tag per imporre il controllo degli accessi? Per prima cosa andiamo nella sezione database e **selezioniamo il nostro database**, creato all'inizio del tutorial. Nelle azioni del database, possiamo selezionare il tag appena creato e il livello di autorizzazione.

Di solito, lasciamo aperto l'accesso al database e limitiamo le autorizzazioni in base alla tabella e ai campi, ma questa scelta di solito varia a seconda del database. Nel nostro esempio, assegniamo il livello public all'intero database di esempio.

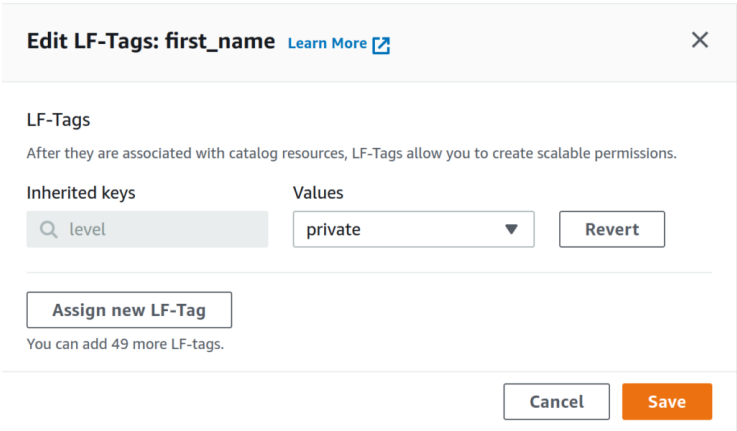


Modifichiamo gli LF-Tag per un intero database

Ora, se vogliamo **restringere l'accesso alle colonne contenenti informazioni personali nella tabella utenti**, possiamo accedere alla tabella da modificare, selezionare una colonna specifica e modificare il suo LF-tag da **public** a **private** (come mostrato nelle figure successive).

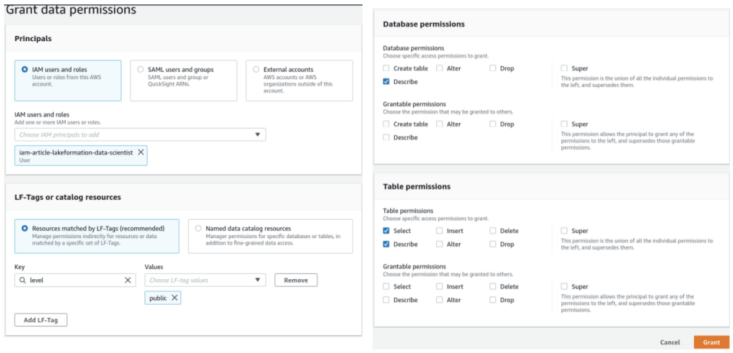


Schema del database di esempio nel quale selezioniamo una colonna



Editiamo gli LF-Tag per i permessi sulle colonne

Ora dobbiamo solo definire quali IAM principals (ad esempio il nostro user di prova) avranno accesso ad uno specifico livello di LF-Tag. Per fare questo, andiamo su *Data lake permissions* e **diamo i permessi ad un utente, ruolo o gruppo IAM accesso alle risoprse taggate con uno specifico LF-Tag**.

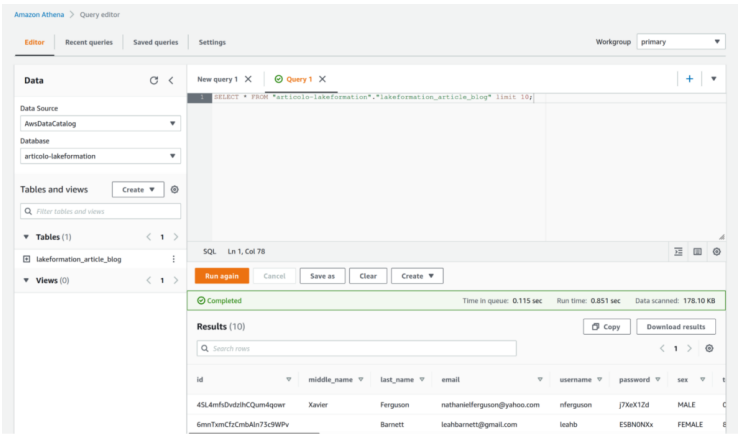


Diamo un'autorizzazione di read

Questo esempio mostra come dare ad un utente l'accesso a tutte le risorse etichettate con "level": "public".

L'utente di esempio potrà così vedere tutti i nostri database ad eccezione dei dati personali etichettati come privati. Un altro utente potrà avere accesso sia alle informazioni pubbliche che a quelle private, aggiungendo il livello *private* nella sezione LF-Tag o, in generale, modificando i tag delle colonne in base alle proprie esigenze.

Ora possiamo interrogare la tabella del database usando il nostro utente di prova che, in base al nostro set di autorizzazioni, non è in grado di vedere la colonna first_name (che è etichettata come privata).



Athena viene usato per fare query ai dati e dimostrare che la colonna first_name non è mostrata nella select perchè taggata come private

Come mostrato nella figura, siamo riusciti a negare al nostro utente di prova il diritto di vedere una colonna "riservata" a nostra scelta.

Vorremmo incoraggiare l'utente a sperimentare aggiungendo o rimuovendo anche le opzioni *describe* e *select* dai permessi LF-Tag nella sezione Data Lake, per vedere come sia possibile negare anche la visualizzazione di interi database o tabelle specifiche.

Nota: a partire dal 3 novembre 2021 per migliorare la sicurezza, AWS Lake Formation ha anche aggiunto il supporto per gli endpoint VPC gestiti tramite AWS PrivateLink per accedere a un data lake in una Virtual Private Cloud.

Feature in preview: sicurezza livello di riga

Lake Formation è ancora un servizio giovane, quindi c'è molto margine di miglioramento. AWS lavora costantemente per aumentare le funzionalità dei suoi servizi e Lake Formation non fa eccezione.

AWS Lake Formation consente già di impostare policy di accesso per nascondere i dati, ad esempio una colonna con informazioni riservate, agli utenti che non dispongono dell'autorizzazione per visualizzare tali dati.

La sicurezza a livello di riga si aggiungerà a ciò, consentendo di impostare criteri a livello di riga oltre ai criteri a livello di colonna.

Un esempio potrebbe essere l'impostazione di una policy che dia a un data scientist l'accesso solo ai dati dell'esperimento contrassegnati con un ID specifico.

Un altro aspetto interessante sarebbe condividere lo stesso Data Lake per set di dati diversi , riducendo così i costi e gli sforzi di gestione.

Concludendo

In questo articolo abbiamo visto come possiamo sfruttare la potenza dei servizi AWS per lo storage e l'analisi dei dati per affrontare la sfida imposta dai Big Data, in particolare come gestire l'accesso, le autorizzazioni e la governance.

Abbiamo dimostrato che i crawler di AWS Glue possono recuperare in modo efficace dati non strutturati da repository temporanei, che si tratti di database come RDS o on-premise, o storage di oggetti come S3, e ottenere uno schema per popolare un catalogo di Glue.

Abbiamo visto che partendo da S3 e da un metadata store è possibile creare un Lake Formation Catalog sopra S3, interamente gestito da AWS, per ridurre drasticamente lo

sforzo di gestione per impostare e amministrare un Data lake.

Abbiamo visto brevemente cos'è una metodologia Tag-Based Access Control (TBAC) e come può essere efficacemente utilizzata per gestire accessi e permessi.

Abbiamo dimostrato che AWS Lake Formation può applicare policy IAM e regole TBAC per concedere o limitare l'accesso a utenti e gruppi anche per riga/colonna. Abbiamo dimostrato che con Lake Formation e AWS Glue, possimao oscurare i dati sensibili a committenti specifici.

Abbiamo descritto gli LF-Tags in dettaglio, con un semplice tutorial. Infine, abbiamo parlato della sicurezza a livello di riga.

Per concludere, possiamo dire che per le sfide relative ai Big Data e alle corrette soluzioni di storage, con un occhio di riguardo alle questioni di sicurezza e governance, le scelte possibili da fare sono sempre due: fai da te o optare per una soluzione gestita.

In questo articolo abbiamo scelto una soluzione gestita per mostrare tutti i vantaggi di un approccio più rigido al problema. Pur essendo una soluzione meno flessibile ad adattarsi, offre un servizio più aderente alle best practices di AWS e meno oneri nell'amministrazione e nella governance.

Come sempre, sentiti libero di commentare nella sezione sottostante e contattaci per qualsiasi dubbio, domanda o idea! Ci vediamo su **Proud2beCloud** tra un paio di settimane per un'altra storia emozionante!

Alessandro Gaggia

Head of software development di beSharp, Full-Stack developer, mi occupo di garantire lo stato dell'arte di tutta la nostra codebase. Scrivo codice in quasi ogni linguaggio, ma prediligo Typescript. Respiro Informatica, Game design, Cinema, Fumetti e buona cucina. Disegno per passione!

Matteo Moroni

DevOps e Solution Architect di beSharp, mi occupo di sviluppare soluzioni Saas, Data Analysis, HPC e di progettare architetture non convenzionali a complessità divergente. Appassionato di informatica e fisica, da sempre lavoro nella prima e ho un PhD nella seconda. Parlare di tutto ciò che è tecnico e nerd mi rende felice!
