

Home > Management & Governance

Lake Formation: Data Security and Data Governance with LF-TBAC

12 November 2021 - 10 min. read



AWS Lake Formation

Data Security and Governance

Database

Introduction

Big Data has rapidly grown as a way to describe information obtained from heterogeneous sources when it becomes incredibly complex to manage in terms of **Variety**, **Veracity**, **Value**, **Volume**, and **Velocity**. Still, it can be considered the "New Gold because of the potential to generate business value."

Without adequate governance or quality, data lakes can quickly turn into unmanageable data swamps. Data engineers know the data they need lives in these swamps, but they won't be able to find, trust, or use it without a clear data governance strategy.

A very common challenge is **maintaining** Governance, **access contro**l over users who operate on the Data Lake, and protecting sensitive information.

Companies need to centralize governance, access control, and a strategy backed by managed services to fine-grain control user access to data.

Dealing with these situations typically requires two approaches: *manual*, **more flexible** but **complex**; *managed* which **requires your solution to fit into specific standards** but in return **takes away all management complexities** for the developers.

This article will guide you through setting up your Data Lake with Lake Formation, showing all the challenges that must be addressed during the process with a particular eye on Security and Governance through the LF-TBAC approach.

Tag-Based Access Control, in short **TBAC**, is an increasingly popular way to solve these challenges, applying constraints based on tags associated with specific resources.

So, without further ado, let's dig in!

What is TBAC access

Tag-based access control allows administrators of IAM-enabled resources to create access policies based on existing tags associated with eligible resources.

Cloud providers manage permissions of both users and applications with policies, documents with rules that reference resources. By applying tags to those resources is possible to define simple and effective allow/deny conditions.

Using access management tags may reduce the number of access policies needed within a cloud account while also providing a simplified way to grant access to a heterogeneous group of resources.

Why S3 alone is not enough

S3, like most AWS services, **leverages the IAM principals for access management**, meaning that it is possible to define which parts of a bucket (files and folders/prefixes) a single IAM principal can read/write; however is not possible to further restrict IAM access to specific parts of an object, nor to certain data segments stored inside objects.

For example, let's assume that our application data is stored as a collection of parquet files divided per country in different folders.

It is possible to constrain a user to access only the users belonging to a given country. Still, there is no way to prevent them from reading the anagraphic information (e.g., username and address) stored as columns in the parquet.

The only way to prevent users from accessing sensitive information would be to encrypt the columns before writing the files to S3, which can be slow, cumbersome, and open a whole new 'can of worm' regarding **key storage**, **sharing**, and eventually **key decommissioning**. Furthermore, giving access to external entities using IAM principals is often a nontrivial problem on its own.

Luckily, AWS offers a **battery included solution to the S3 Data Lake permission problem**: enters AWS Lake Formation!

AWS Lake Formation is a fully managed service that simplifies building, securing, and managing data lakes, automating many of the complex manual steps required to create them.

Lake Formation also provides its own permissions model, which is what we want to explore in detail, that augments the classical AWS IAM permissions model.

This centrally defined permissions model enables fine-grained access to data stored in data lakes through a simple grant/revoke mechanism.

So, by leveraging the power of Lake Formation, we would like to demonstrate, with a simple solution, how to address the aforementioned S3 challenges; let's continue!

Leveraging TBAC approach in Lake Formation

To accompany the reader in understanding why AWS Lake Formation can be a good choice in dealing with the complexities of managing a DataLake, we have prepared a simple tutorial on how to migrate heterogeneous data.

From legacy on-prem databases into S3 while also creating a Lake Formation catalog to deal with data cleansing, permissions, and further operations.



Our example implementation

AWS Glue migration of on-prem data

The first step for creating a Data Lake is obviously to fetch, transform and insert the data. In this simple example, we used a mocked users dataset from a MySQL database. AWS glue is the natural way to connect to the heterogeneous data source, infer their schema import and transform the data and finally write them on S3 as we explained in detail here.

After the data is loaded in a temporary S3 bucket, you need to create a **Database in Lake Formation** to connect to a **Glue Crawler** and run it on your S3 prefix to populate a Glue Catalog for your data.

Just go to the **AWS Lake Formation console,** in the *Databases* page under the **Data catalog tab,** and fill in a Database name and your S3 path.

Database deta Treate a database in t	ils e AWS Glue Data Catalog.
• Database Create a datab	se in my account. Resource link Create a resource link to a shared database.
lame	
articolo-lakeforn	
ocation - optiona	ation
ocation - optional ihoose an Amazon S: his location's childrei s3://lakeformati	path for this database, which eliminates the need to grant data location permissions on catalog table paths that are n-article-blog Browse
ocation - optional hoose an Amazon S: his location's childrer s3://lakeformati Description - optio	path for this database, which eliminates the need to grant data location permissions on catalog table paths that are n-article-blog Browse al
ocation - optional ihoose an Amazon S: his location's childrei s3://lakeformati	path for this database, which eliminates the need to grant data location permissions on catalog table paths that are n-article-blog Browse al
ocation - optiona hoose an Amazon S3 hoose on Amazon S3 s3://lakeformati Description - optio articolo-lakeform	path for this database, which eliminates the need to grant data location permissions on catalog table paths that are n-article-blog Browse al

Create a new Database from Lake Formation

Note: creating a database from Lake Formation assures correct permissions are associated with it, we could have done the same thing from AWS Glue but we would have needed extra effort to modify permissions for the next steps.

After the database is created, we need the Glue Catalog, which is a metastore containing the schema (schema-on-read) of your data saved in S3 (usually as parquet files).

Having a Glue Schema is necessary to set up the AWS Lake Formation access layer in

front of your S3 Data Lake. To make it, just create a Crawler and link it to the same S3 path as the Database, and **set that DB as the crawler output**.

Crawler info	Add a data store
article-lakeformation- crawler	Choose a data store
Crawler source type	83
Data stores Data store	Connection
	Select a connection
IAM Role Schedule	Optionally include a Network connection to use with this S3 target. Note that each orawier is limited to one Network connection so any future targets will also use the same connection (or none, if left blank).
Output	Add connection
	Crawl data in
	Specified path in my account Specified path in another account
	Include path
	s3://lakeformation-article-blog
	All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder to crawl all objects in MyFolder v MyBucket.
	Sample size (optional)
	Enter an integer between 1 and 249.
	This field sets the number of files in each leaf folder to be crawled. If not set, all the files are crawled.
	 Exclude patterns (optional)
	Back Next

Setup of a basic AWS Glue Crawler

In order to use the Crawler, an IAM role is necessary, but luckily AWS has a step for that in the Crawler creation wizard:

	Choose an IAM role							
 Update a policy in a Choose an existing Create an IAM role 								
IAM role 🚯								
AWSGlueServiceRole-	lake-formation-article							
s3://lakeformation-article-blog								
	Back Next							

How to create an IAM role for using the Crawler

Once the Crawler is created, and data is imported into the catalog, we are ready for the next step.

00	events		
		and match terms, phrases, or values in your log events. Learn more about filter patterns 🕑	□ View as text C Actions ▼ Create Metric Filter
Q	"fd4bad47-4512-4d7b-ab13-d76ff6ddf	561' >	Clear 1m 30m 1h 12h Custom 🗐 🕻
•	Timestamp	Message	
•	2021-10-18T16:53:39.742+02:00	[fd4bad47-4512-4d7b-ab13-d76ff6ddf561] BENCHMARX : Running Start Crawl for Crawler articl	le-lakeformation-crawler
•	2021-10-18T16:53:57.810+02:00	[fd4bad47-4512-4d7b-ab13-d76ff6ddf561] BENCHMARK : Classification complete, writing result	lts to database articolo-lakeformation
۱.	2021-10-18T16:53:57.812+02:00	[fd4bad47-4512-4d7b-ab13-d76ff6ddf561] INFO : Crawler configured with SchemaChangePolicy	{"UpdateBehavior":"UPDATE_IN_DATABASE","DeleteBehavior":"DELETE
•	2021-10-18T16:54:19.821+02:00	[fd4bad47-4512-4d7b-ab13-d76ff6ddf561] INFO : Created table lakeformation_article_blog in	n database articolo-lakeformation
•	2021-10-18T16:54:21.748+02:00	[fd4bad47-4512-4d7b-ab13-d76ff6ddf561] BENCHMARX : Finished writing to Catalog	
•	2021-10-18T16:55:28.892+02:00	[fd4bad47-4512-4d7b-ab13-d76ff6ddf561] BENCHMARK : Crawler has finished running and is in	n state READY

Cloudwatch Logs demonstrating that Crawler worked correctly

AWS Lake Formation

By having a Glue Data catalog in place, it is time to set up Lake Formation to finally manage user access permissions.

In order to do so, let's start by going to the Lake Formation dashboard and **removing the usual S3 access permissions**.

Dashboard Data catalog Databases		abases (0/2) Find dotabases					C Actions V	View tabl	Create database
Tables		Name	Owner account ID	Ŧ	Shared resource	Ŧ	Shared resource owner	v	Amazon 53 path
Settings		articolo-lakeformation	364050767034						s3://lakeformation-article-bi
Register and ingest Data lake locations		memo-test-banor	364050767034						
Blueprints	_							_	
Crawlers 🖸									
Johs 🗹									
Permissions									
Administrative roles and tasks									
LF-Tags									
LF-tag permissions									
Data lake permissions									
Data locations									
External data filtering									

Lake Formation dashboard

So we can go to *Data Catalog Settings* and uncheck *Use only IAM access control for new databases* and *Use only IAM access control for new tables in new databases.* By default, access to Data Catalog resources and Amazon S3 locations are controlled solely by AWS Identity and Access Management (IAM) policies, unchecking the values allows Individual Lake Formation permissions to take effect.

•	rmissions alert	
	e "Use only IAM access control" settings below are enabled for compatibility with existing Glue Data Catalog behavior. We re <u>/S Glue Data Permissions to the Lake Formation Model [2]</u>	ommend that you read the documentation on <u>Changing Default Settings for Your Data Lake [2]</u> and Upgra
	IS Lake Formation > Data catalog settings	
)	ata catalog settings	
	Default permissions for newly created databases and tables	
	Default permissions for newly created databases and tables	
	These settings maintain existing AWS Glav Data Catalog behavior. You can still set individual permissions on databases and tables, which will take effect when you revoke the Super permission from IAMAllowe@Principals. See Changing Default Settings for Your Data Lake .	
	Use only IAM access control for new databases	
	Use only IAM access control for new tables in new databases	
	Default permissions for AWS CloudTrail These settings specify the information being shown in AWS CloudTrail.	
	Resource owners Enter resource owners you wish to share your CloudThal access details with.	
	Q. Enter an AWS account ID	
	Enter one or more AWS account IDs. Press Enter after each ID.	

Lake Formation data catalog setting: disable both the Use only flag

Once access **responsibilities are delegated to Lake Formation**, we can remove the access for the standard IAMAllowedPrincipals IAM group, in the data lake *Permissions* tab, select the **permission of the IAM group** and click *Revoke*.

Dashboard Data catalog Databases		permissions for data		eformation (2)					G	Revoke Grant
Tables Settings 🕢		Principal	Principal type $ \nabla $	Resource type $ \nabla $	Database v	Table ∇	Resource v	Catalog $ au$	LF-tag expressions	Permissions
Register and ingest	0	IAMAllowedPrincipals	Group	Database	articolo-lakeformation		articolo-lakeformation	364050767034		AL
Data lake locations		matteo.moroni	IAM role	Database	articolo-lakeformation		articolo-lakeformation	364050767034		All, Alter, Create table, Desci
Blueprints Crawlers [2] Jobs [2] Permissions Administrative roles and tasks (J-Tags LF-tags permissions Data lake permissions										

Revoke standard IAMAllowedPrincipals permissions

The user creating the DataLake will also be listed in this section with admin privileges, if you want that user to retain access to the data you can leave the permission as they are, otherwise you can either **revoke the permission to the user/role or restrict them**. *Note: if you need to add a Data lake administrator principal, you can do so by going to the Administrative roles and tasks and adding a Data lake admin.*

	▼ How it works		
ita catalog tabinses bites titings inglister and ingest ta lake locations upprints	1 Set administrative roles Decide who should be the administration for your data lake, and optionally who can oracle new disbases. Obesise administratives	2 Define UF-tag endology In order to create and manage catalog and data access permasane, drive a set of UT-tags that will help you gatchly decide all your of access needs. Manage LP-Tags	Delegate U-Lag permissions - eptianul Latity, you can decide who should see the U-Lag ontoingy and U-Lag catalog resources (disbubanes, tables, columna) in order to each cleas Manage U-Lag permissions.
awiers 🙆 to 🖾 ministrative roles and tasks UF-Tags UF-tag permissions	Data take administrators (0/2) Administrators can view all installes in the ANS Gale Data Cables, They can also grain Q. /red administrators		C Choose administrators
ta lake permissions	Name	 Туре 	
ita locations ternal data filtering	beSharp-admin matteo.moroni	IAM role	
ternal data mitering		SANI INRE	
	Database creators		C Revoke Grant
	Database creators Doese VM principals permitted to create databases in your XMS Give Data Catalog Q, Faind distributor creators		C Revoke Grant

Add admin and db creator console

Once all these steps are completed, it is time to start defining Lake Formation tags (**LF-Tags** from now on), which will be used to restrict access to the data lake. From the *LF-Tags* page under the *Permissions* tab **create a new LF-Tag** and for key use *level* and add *private, sensitive,* and *public* as value separated by comma just like in the figure. Click **Add LF-tag**.

Add LF-Tag Learn More	×
LF-Tags have a key and one or more values that can be associated with data catalog resources. T automatically inherit from database LF-tags, and columns inherit from table LF-tags. Example: Key = Confidentiality Values = private, sensitive, public Key	ables
level	
Key string must be less than 128 characters long, and cannot be changed once LF-tag is created	
Values Type a single value and select [Enter] or specify multiple values separated by commas.	
Add	
private X sensitive X public X Enter up to 15 values; each value must be less than 256 characters long.	
Cancel Add LF-1	tag

LF-Tag creation

Now once created, how can we use these tags to enforce access control? First of all, let's go to the database section and **select our database**, created at the beginning of the tutorial. In *database actions*, you can select the tag you've created and the permission level.

Usually, we leave the database access open and restrict permissions on a per table and fields basis, but this is different for each database. In our example, we assign the level **public** to the whole example database.

Edit LF-Tags: articolo-lakeformation Learn More 🔀								
LF-Tags After they are associated with catalog Assigned keys	resources, LF-Tags allow you to create so	alable permissions.						
Q level X	public	Remove						
Assign new LF-Tag You can add 49 more LF-tags.								
	Ca	ncel Save						

Edit LF-Tag for the entire database

Now if we want to **restrict access to the columns in the user table containing personal info**, we can go to the table to modify, select the column and change its LF-tag from **public** to **private** (see figures).

Sche	ema Find Columns		lete Edit	Edit tags	Add column
	#	Column Name 🗸 🗸	Data ty ⊽	Partition key	Comment
-	# 1	id	string	-	-
	2	first_name	string	-	-
	3	middle_name	string	-	-
	4	last_name	string	-	-
	5	email	string	-	-
	6	username	string	-	-
	7	password	string	-	-
	8	sex	string	-	-
	9	telephone_number	string	-	-
	10	date_of_birth	string	-	-
	11	age	double	-	-
	12	company_email	string	-	-
	13	national_identity_card_number	string	-	-
	14	national_identification_num	double	-	-
	15	passport_number	string	-	-
	16	bank_account	string	-	-

Schema of our example database in which we select a column

Edit LF-Tags: first_name	Learn More	×
,	resources, LF-Tags allow you to create sc	alable permissions.
Q level	Values private	Revert
Assign new LF-Tag You can add 49 more LF-tags.	Ca	ncel Save

Editing a per column LF-Tag

Now we just need to define which IAM principals (i.e, our test user) will have access to a given LF-Tag. To do so, let's go to *Data lake permissions* and **grant permissions to an IAM user/role/group to access resources tagged with a given LF-Tag**.

Principals				
rincipats		Database permissions Choose specific access permissions	o grant.	
Users or roles from this AWS SAML users and group or AWS	mal accounts accounts or AINS nizations outside of this art.	Create table Alter	Drop	Super This permission is the union of all the individual permissions to the left, and supersedes them.
M users and roles		Grantable permissions Choose the permission that may be	pranted to others.	
idd one or more IAM users or roles.		Create table Alter	Drop	Super
Choose IAM principals to add		Describe		This permission allows the principal to grant any of the permissions to the left, and supersedes those grantable permissions.
				permissions.
Uuer		Table permissions Table permissions	n arant.	permane.
F-Tags or catalog resources Resources Manual data catalog resources	NIFCES		o grant.	permanen.
Imma at the lateformation data scients: X F-Tags or catalog resources Fass without 9/LF-Tags becomeaded Record provide and of 2-fass, C Imma data catalog resources to data Imma data catalog resources Imma data data data data Imma data data data data Imma data data data data Imma data data data Imma data data data Imma data data	cific databases or tables, in	Table permissions Choose specific access permissions		
F-Tags or catalog resources For any of the second	cific databases or tables, in access.	Table permissions Choose specific access permissions f Select Insert	Delete	Soper The permission is the union of all the individual correlations
T-Tags or catalog resources Frage or catalog resources Measures matched by US-Tags tecomended: Manuel data catalog new Manuel data catalog new Manuel data catalog new matched by users of an	cific databases or tables, in	Table permissions Deces specific access permissions Select insert Describe Alter Grantable permissions	Delete	Soper The permission is the union of all the individual correlations

Grant read permission

This example shows how to give a user access to all the resources tagged with "level": "public".

This user will thus be able to see all our databases except for the personal data tagged as private. Another user may have access to both public and private information, just add the private level in the LF-Tag section or modify columns tags according to your needs.

We can now query the database table using our test user which, based on our set of permissions, is not able to see the first_name column (which is tagged as private).



Athena is used to querying data and demonstrating that first_name is not shown in the table select because is tagged as private

As shown in the figure we have successfully managed to deny our test user the right to see a "sensible" column of our choice.

We would like to encourage the user to experiment in adding or removing also describe and select options from the LF-Tag permissions in the Data Lake section to see that we can also deny listing both database and tables.

Note: as of Nov 3, 2021: to enhance security, AWS Lake Formation also added support for managed VPC endpoints via AWS PrivateLink to access a data lake in a Virtual Private Cloud.

Feature in preview: row-level security

Lake Formation is still a young service, so there is much room for improvement. AWS is constantly working on increasing features for its services, and Lake Formation is no exception.

AWS Lake Formation already allows setting access policies to hide data, such as a column with sensitive information, from users who do not have permission to view that data.

Row-level security will add up to that by allowing to set row-level policies in addition to column-level policies.

An example could be setting a policy that gives a data scientist access to only the experiment data marked with a specific id.

Another interesting aspect would be to share the same Data Lake for different datasets to reduce costs and management efforts.

To Sum up

In this article, we have seen how we can leverage the power of AWS Services for Storage and Data Analytics to tackle the challenge imposed by Big Data, in particular how to manage access, permissions, and governance.

We have shown that AWS Glue crawlers can effectively retrieve unstructured data from temporary repositories, being them databases like RDS or on-premises, or object storages like S3, and obtain a schema to populate a Glue Catalog.

We have seen that starting from S3 and a metadata store, it is possible to create a Lake Formation Catalog on top of S3, entirely managed by AWS, to drastically reduce the management effort to set up and administrate a Data lake.

We have briefly seen what is a Tag-Based Access Control (TBAC) methodology and how can be effectively used to manage access and permissions. We have shown that AWS Lake Formation can apply IAM policies and TBAC rules to give or restrain access to users and groups even on a per-column/row basis. We demonstrated that with Lake Formation and AWS Glue, we could obscure sensitive data to specific principals.

We have described LF-Tags in detail, with a simple tutorial. Finally, We have talked about Row-Level Security.

To conclude, we can say that for challenges regarding Big Data and proper storage solutions, with an eye for security and governance matters, there are always two possible choices to make: DIY or opt for a managed solution.

In this article, we chose a **managed** solution to show all the benefits of a more rigid approach to the problem. Despite being less flexible to adaptation, it offers a service more adherent to best practices and less burden in administration and governance.

As always, feel free to comment in the section below, and reach us for any doubt, question or idea!

See you on **Proud2beCloud** in a couple of weeks for a new story!



Alessandro Gaggia

Head of software development at beSharp and Full-Stack Developer, I keep all our codebases up-to-date. I write code in almost any language, but Typescript is my favorite. I live for IT, Game design, Cinema, Comics, and... good food. Drawing is my passion!



Matteo Moroni

DevOps and Solution Architect at beSharp, I deal with developing Saas, Data Analysis, and HPC solutions, and with the design of unconventional architectures with different

complexity. Passionate about computer science and physics, I have always worked in the first and I have a PhD in the second. Talking about anything technical and nerdy makes me happy!

Copyright © 2011-2021 by beSharp srl - P.IVA IT02415160189