Design a data mesh architecture using AWS Lake Formation and AWS Glue

by Nivas Shankar, Ian Meyers, Zach Mitchell, and Roy Hasson | on 09 JUL 2021 | in Analytics, AWS Big Data, AWS Glue, AWS Lake Formation, Serverless | Permalink | Comments | Share

Organizations of all sizes have recognized that data is one of the key enablers to increase and sustain innovation, and drive value for their customers and business units. They are eagerly modernizing traditional data platforms with cloud-native technologies that are highly scalable, feature-rich, and cost-effective. As you look to make business decisions driven by data, you can be agile and productive by adopting a mindset that delivers data products from specialized teams, rather than through a centralized data management platform that provides generalized analytics.

In this post, we describe an approach to implement a data mesh using AWS native services, including AWS Lake Formation and AWS Glue. This approach enables lines of business (LOBs) and organizational units to operate autonomously by owning their data products end to end, while providing central data discovery, governance, and auditing for the organization at large, to ensure data privacy and compliance.

Benefits of a data mesh model

A centralized model is intended to simplify staffing and training by centralizing data and technical expertise in a single place, to reduce technical debt by managing a single data platform, and to reduce operational costs. Data platform groups, often part of central IT, are divided into teams based on the technical functions of the platform they support. For instance, one team may own the ingestion technologies used to collect data from numerous data sources managed by other teams and LOBs. A different team might own data pipelines, writing and debugging extract, transform, and load (ETL) code and orchestrating job runs, while validating and fixing data quality issues and ensuring data processing meets business SLAs. However, managing data through a central data platform can create scaling, ownership, and accountability challenges, because central teams may not understand the specific needs of a data domain, whether due to data types and storage, security, data catalog requirements, or specific technologies needed for data processing.

You can often reduce these challenges by giving ownership and autonomy to the team who owns the data, best allowing them to build data products, rather than only being able to use a common central data platform. For instance, product teams are responsible for ensuring the product inventory is updated regularly with new products and changes to existing ones. They're the domain experts of the product inventory datasets. If a discrepancy occurs, they're the only group who knows how to fix it. Therefore, they're best able to implement and operate a technical solution to ingest, process, and produce the product inventory dataset. They own everything leading up to the data being consumed: they choose the technology stack, operate in the mindset of data as a product, enforce security and auditing, and provide a mechanism to expose the data to the organization in an easy-to-consume way. This reduces overall friction for information flow in the organization, where the producer is responsible for the datasets they produce and is accountable to the consumer based on the advertised SLAs.

This data-as-a-product paradigm is similar to Amazon's operating model of building services. Service teams build their services, expose APIs with advertised SLAs, operate their services, and own the end-to-end customer experience. This is distinct from the world where someone builds the software, and a different team operates it. The end-to-end ownership model has enabled us to implement faster, with better efficiency, and to quickly scale to meet customers' use cases. We aren't limited by centralized teams and their ability to scale to meet the demands of the business. Each service we build stands on the shoulders of other services that provide the building blocks. The analogy in the data world would be the data producers owning the end-to-end implementation and serving of data products, using the technologies they selected based on their unique needs. At AWS, we have been talking about the data-driven organization model for years, which consists of data producers and consumers. This model is similar to those used by some of our customers, and has been eloquently described recently by Zhamak Dehghani of Thoughtworks, who coined the term data mesh in 2019.

Solution overview

In this post, we demonstrate how the Lake House Architecture is ideally suited to help teams build data domains, and how you can use the data mesh approach to bring domains together to enable data sharing and federation across business units. This approach can enable better autonomy and a faster pace of innovation, while building on top of a proven and well-understood architecture and technology stack, and ensuring high standards for data security and governance.

The following are key points when considering a data mesh design:

- Data mesh is a pattern for defining how organizations can organize around data domains with a focus on delivering data as a product. However, it may not be the right pattern for every customer.
- A Lake House approach and the data lake architecture provide technical guidance and solutions for building a modern data platform on AWS.
- The Lake House approach with a foundational data lake serves as a repeatable blueprint for implementing data domains and products in a scalable way.
- The manner in which you utilize AWS analytics services in a data mesh pattern may change over time, but still remains consistent with the technological recommendations and best practices for each service.

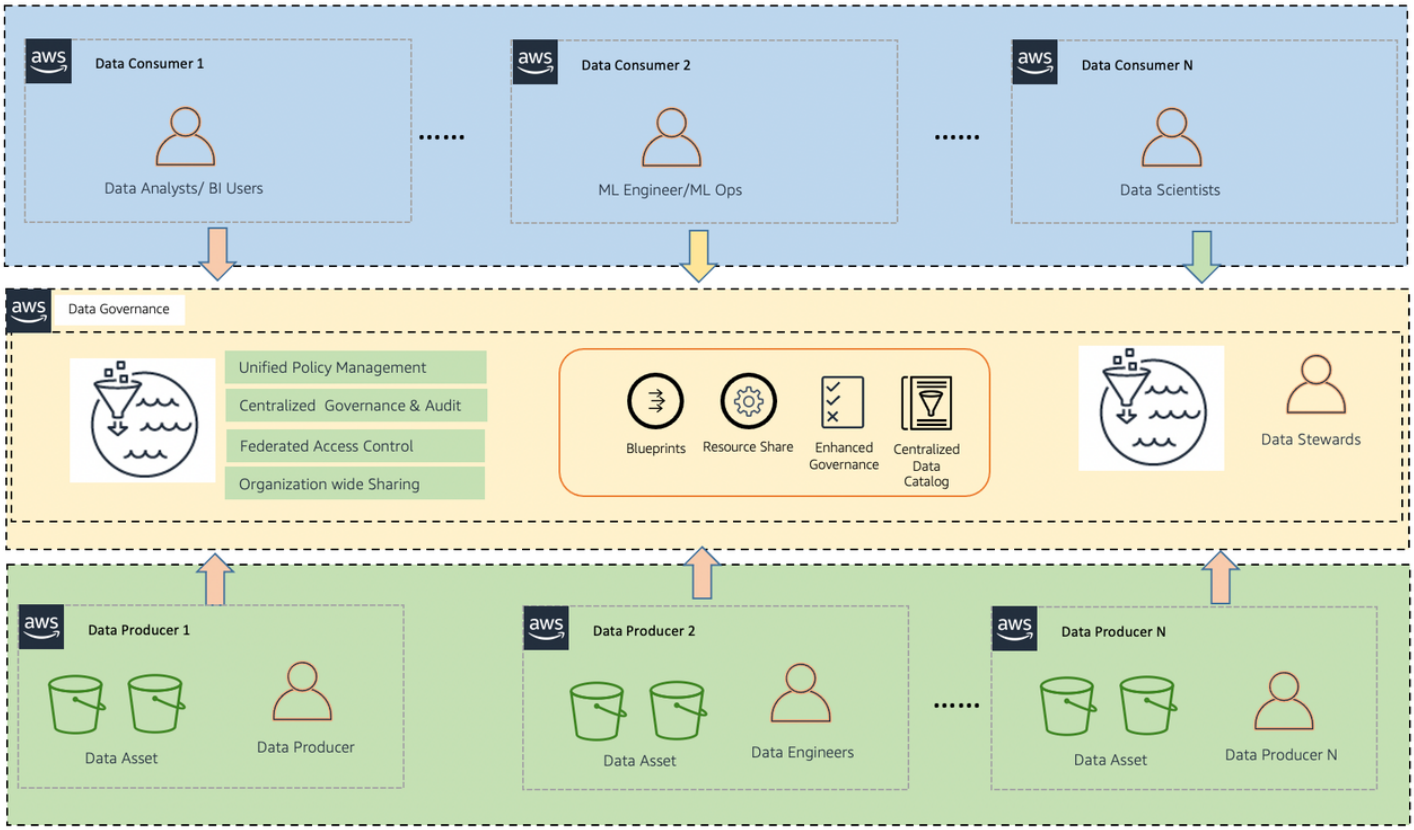
The following are data mesh design goals:

- **Data as a product** – Each organizational domain owns their data end to end. They're responsible for building, operating, serving, and resolving any issues arising from the use of their data. Data accuracy and accountability lies with the data owner within the domain.
- **Federated data governance** – Data governance ensures data is secure, accurate, and not misused. The technical implementation of data governance such as collecting lineage, validating data quality, encrypting data at rest and in transit, and enforcing appropriate access controls can be managed by each of the data domains. However, central data discovery, reporting, and auditing is needed to make it simple for users to find data and for auditors to verify compliance.
- **Common Access** – Data must be easily consumable by subject matter personas like data analysts and data scientists, as well as purpose-built analytics and machine learning (ML) services like Amazon Athena, Amazon Redshift, and Amazon SageMaker. To do that, data domains must expose a set of interfaces that make data consumable while enforcing appropriate access controls and audit tracking.

The following are user experience considerations:

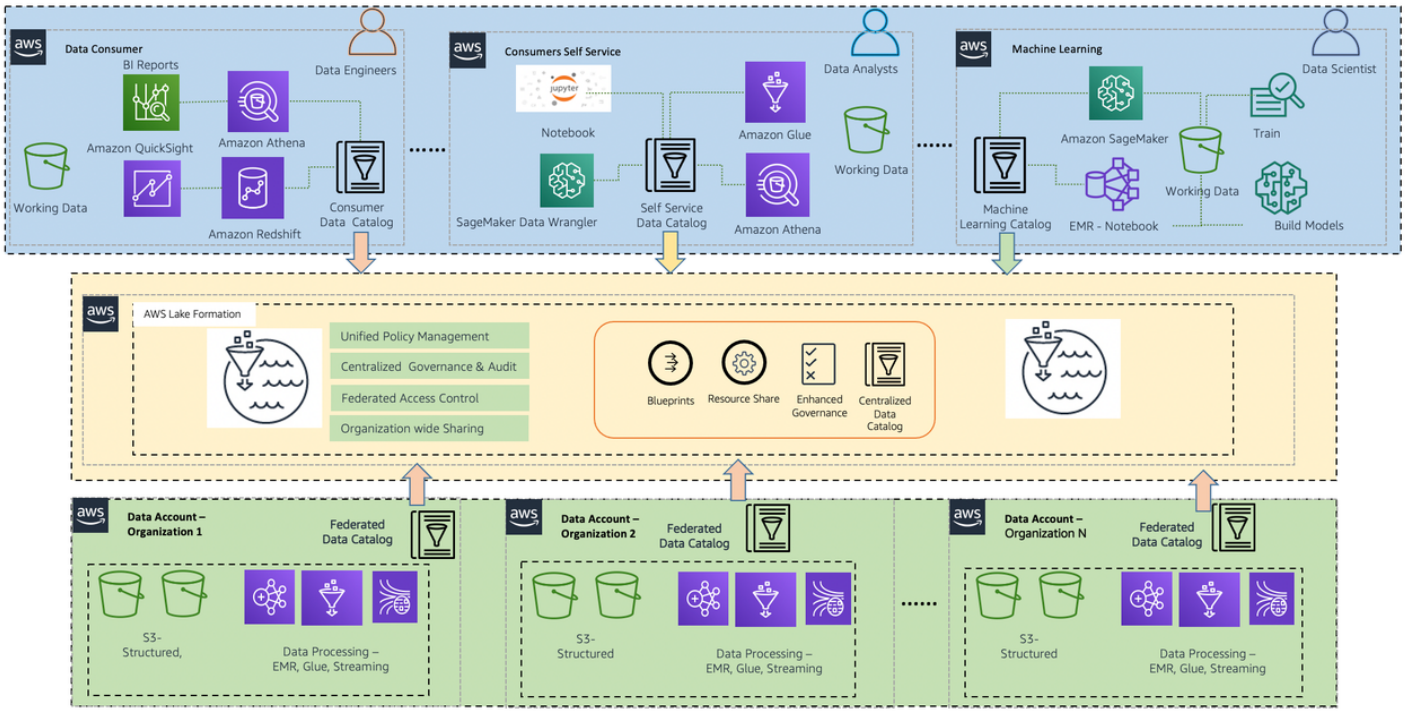
- Data teams own their information lifecycle, from the application that creates the original data, through to the analytics systems that extract and create business reports and predictions. Through this lifecycle, they own the data model, and determine which datasets are suitable for publication to consumers.
- Data domain producers expose datasets to the rest of the organization by registering them with a central catalog. They can choose what to share, for how long, and how consumers can interact with it. They're also responsible for maintaining the data and making sure it's accurate and current.
- Data domain consumers or individual users should be given access to data through a supported interface, like a data API, that can ensure consistent performance, tracking, and access controls.
- All data assets are easily discoverable from a single central data catalog. The data catalog contains the datasets registered by data domain producers, including supporting metadata such as lineage, data quality metrics, ownership information, and business context.
- All actions taken with data, usage patterns, data transformation, and data classifications should be accessible through a single, central place. Data owners, administrators, and auditors should able to inspect a company's data compliance posture in a single place.

Let's start with a high-level design that builds on top of the data mesh pattern. As seen in the following diagram, it separates consumers, producers, and central governance to highlight the key aspects discussed previously. However, a data domain may represent a data consumer, a data producer, or both.



The objective for this design is to create a foundation for building data platforms at scale, supporting the objectives of data producers and consumers with strong and consistent governance. The AWS approach to designing a data mesh identifies a set of general design principles and services to facilitate best practices for building scalable data platforms, ubiquitous data sharing, and enable self-service analytics on AWS.

Expanding on the preceding diagram, we provide additional details to show how AWS native services support producers, consumers, and governance. Each data domain, whether a producer, consumer, or both, is responsible for its own technology stack. However, using AWS native analytics services with the Lake House Architecture offers a repeatable blueprint that your organization can use as you scale your data mesh design. Having a consistent technical foundation ensures services are well integrated, core features are supported, scale and performance are baked in, and costs remain low.

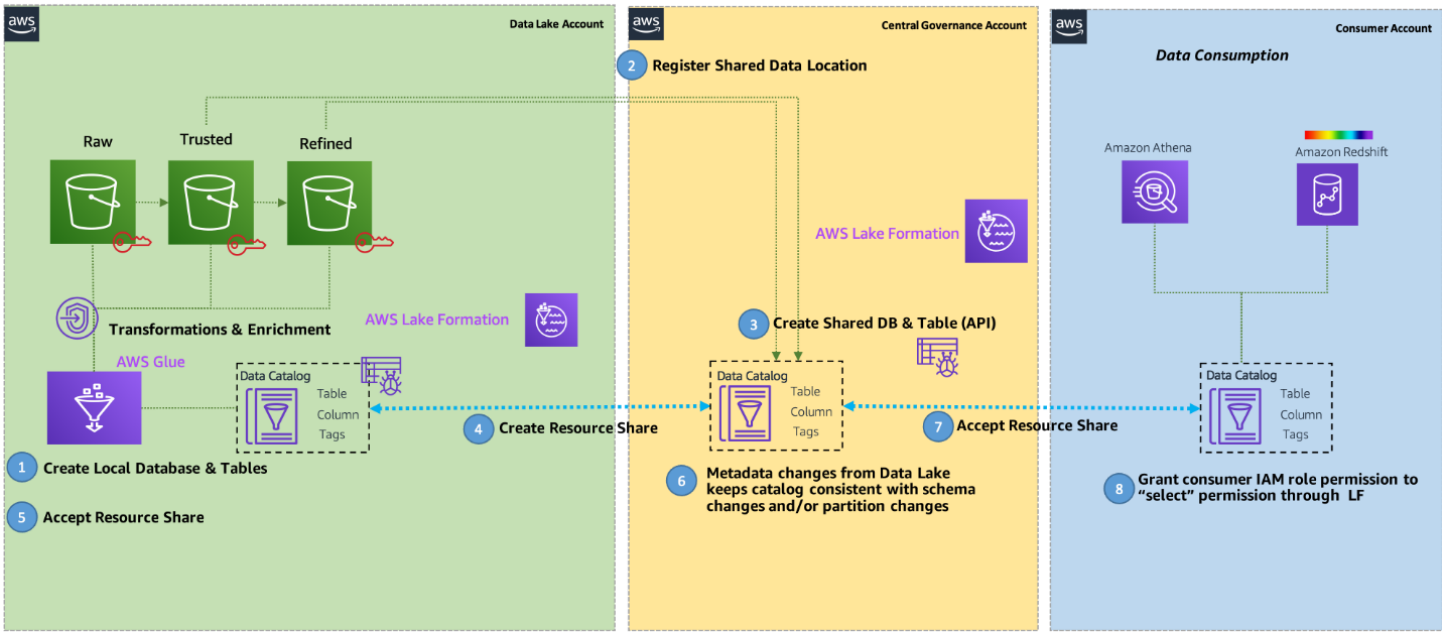


A data domain: producer and consumer

A data mesh design organizes around data domains. Each data domain owns and operates multiple data products with its own data and technology stack, which is independent from others. Data domains can be purely producers, such as a finance domain that only produces sales and revenue data for domains to consumers, or a consumer domain, such as a product recommendation service that consumes data from other domains to create the product recommendations displayed on an ecommerce website. In addition to sharing, a centralized data catalog can provide users with the ability to more quickly find available datasets, and allows data owners to assign access permissions and audit usage across business units.

A producer domain resides in an AWS account and uses Amazon Simple Storage Service (Amazon S3) buckets to store raw and transformed data. It maintains its own ETL stack using AWS Glue to process and prepare the data before being cataloged into a Lake Formation Data Catalog in their own account. Similarly, the consumer domain includes its own set of tools to perform analytics and ML in a separate AWS account. The central data governance account is used to share datasets securely between producers and consumers. It's important to note that sharing is done through metadata linking alone. Data isn't copied to the central account, and ownership remains with the producer. The central catalog makes it easy for any user to find data and to ask the data owner for access in a single place. They can then use their tool of choice inside of their own environment to perform analytics and ML on the data.

The following diagram illustrates the end-to-end workflow.



The workflow from producer to consumer includes the following steps:

1. Data source locations hosted by the producer are created within the producer's AWS Glue Data Catalog and registered with Lake Formation.
2. When a dataset is presented as a product, producers create Lake Formation Data Catalog entities (database, table, columns, attributes) within the central governance account. This makes it easy to find and discover catalogs across consumers. However, this doesn't grant any permission rights to catalogs or data to all accounts or consumers, and all grants are be authorized by the producer.
3. The central Lake Formation Data Catalog shares the Data Catalog resources back to the producer account with required permissions via Lake Formation resource links to metadata databases and tables.
4. Lake Formation permissions are granted in the central account to producer role personas (such as the data engineer role) to manage schema changes and perform data transformations (alter, delete, update) on the central Data Catalog.
5. Producers accept the resource share from the central governance account so they can make changes to the schema at a later time.
6. Data changes made within the producer account are automatically propagated into the central governance copy of the catalog.
7. Based on a consumer access request, and the need to make data visible in the consumer's AWS Glue Data Catalog, the central account owner grants Lake Formation permissions to a consumer account based on direct entity sharing, or based on tag based access controls, which can be used to administer access via controls like data classification, cost center, or environment.
8. Lake Formation in the consumer account can define access permissions on these datasets for local users to consume. Users in the consumer account, like data analysts and data scientists, can query data using their chosen tool such as Athena and Amazon Redshift.

Build data products

Data domain producers ingest data into their respective S3 buckets through a set of pipelines that they manage, own, and operate. Producers are responsible for the full lifecycle of the data under their control, and for moving data from raw data captured from applications to a form that is suitable for consumption by external parties. AWS Glue is a serverless data integration and preparation service that offers all the components needed to develop, automate, and manage data pipelines at scale, and in a cost-effective way. It provides a simple-to-use interface that organizations can use to quickly onboard data domains without needing to test, approve, and juggle vendor roadmaps to ensure all required features and integrations are available.

Central data governance

The central data governance account stores a data catalog of all enterprise data across accounts, and provides features allowing producers to register and create catalog entries with AWS Glue from all their S3 buckets. No data (except logs) exists in this account. Lake Formation centrally defines security, governance, and auditing policies in one place, enforces those policies for consumers across analytics applications, and only provides authorization and session token access for data sources to the role that is requesting access. Lake Formation also provides uniform access control for enterprise-wide data sharing through resource shares with centralized governance and auditing.

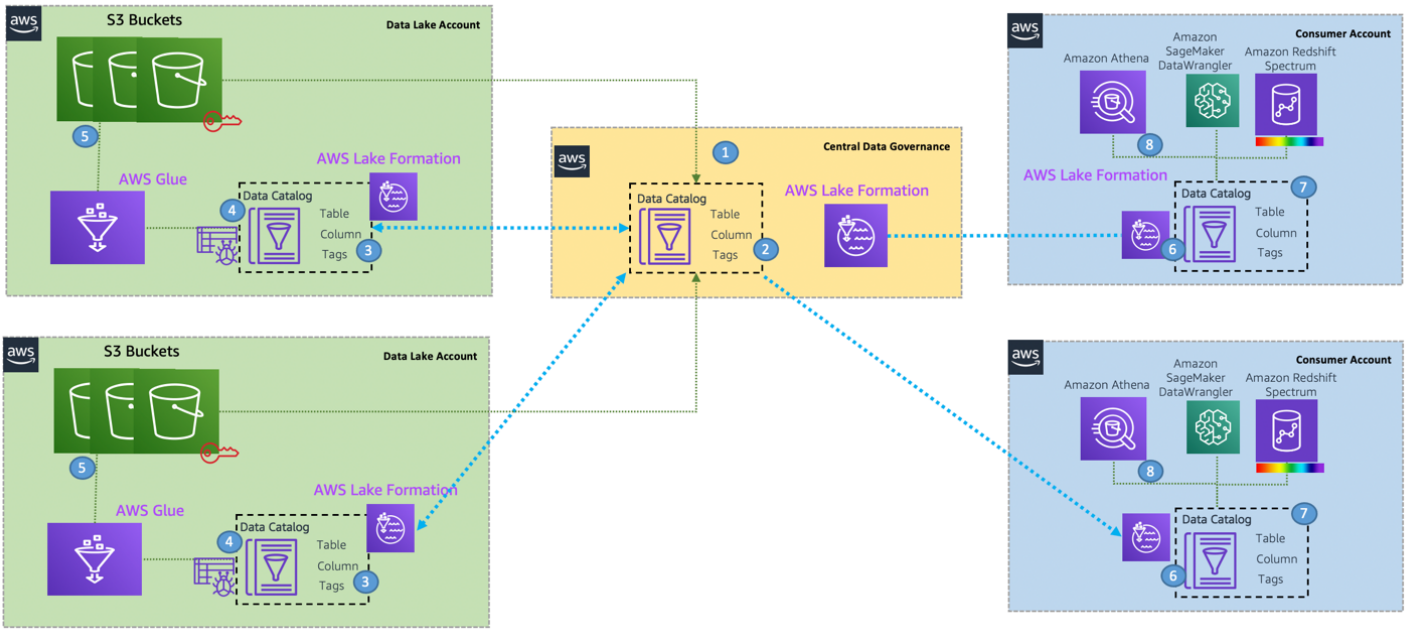
Common access

Each consumer obtains access to shared resources from the central governance account in the form of resource links. These are available in the consumer's local Lake Formation and AWS Glue Data Catalog, allowing database and table access that can be managed by consumer admins. After access is granted, consumers can access the account and perform different actions with the following services:

- Athena acts as a consumer and runs queries on data registered using Lake Formation. Lake Formation verifies that the workgroup AWS Identity and Access Management (IAM) role principal has the appropriate Lake Formation permissions to the database, table, and Amazon S3 location as appropriate for the query. If the principal has access, Lake Formation vends temporary credentials to Athena, and the query runs. Authentication is granted through IAM roles or users, or web federated identities using SAML or OIDC. For more information, see How Athena Accesses Data Registered With Lake Formation.
- Amazon SageMaker Data Wrangler allows you to quickly select data from multiple data sources, such as Amazon S3, Athena, Amazon Redshift, Lake Formation, and Amazon SageMaker Feature Store. You can also write queries for data sources and import data directly into SageMaker from various file formats, such as CSV files, Parquet files, and database tables. Authentication is granted through IAM roles in the consumer account. For more information, see Prepare ML Data with Amazon SageMaker Data Wrangler.
- Amazon Redshift Spectrum allows you to register external schemas from Lake Formation, and provides a hierarchy of permissions to control access to Amazon Redshift databases and tables in a Data Catalog. If the consumer principal has access, Lake Formation vends temporary credentials to Redshift Spectrum tables, and the query runs. Authentication is granted through IAM roles or users, or web federated identities using SAML or OIDC. For more information, see Using Redshift Spectrum with AWS Lake Formation.
- Amazon QuickSight via Athena integrates with Lake Formation permissions. If you're querying data with Athena, you can use Lake Formation to simplify how you secure and connect to your data from QuickSight. Lake Formation adds to the IAM permissions model by providing its own permissions model that is applied to AWS analytics and ML services. Authentication is granted through IAM roles that are mapped to QuickSight user permissions. For more information, see Authorizing Connections Through AWS Lake Formation.

- Amazon EMR Studio and EMR notebooks allow running Spark SQL against Lake Formation's tables backed by a SAML authority. Beginning with Amazon EMR 5.31.0, you can launch a cluster that integrates with Lake Formation. Authentication is granted through IAM roles or users, or web federated identities using SAML or OIDC. For more information, see Integrate Amazon EMR with AWS Lake Formation.

With this design, you can connect multiple data lake houses to a centralized governance account that stores all the metadata from each environment. The strength of this approach is that it integrates all the metadata and stores it in one meta model schema that can be easily accessed through AWS services for various consumers. You can extend this architecture to register new data lake catalogs and share resources across consumer accounts. The following diagram illustrates a cross-account data mesh architecture.



Conclusion

A data mesh approach provides a method by which organizations can share data across business units. Each domain is responsible for the ingestion, processing, and serving of their data. They are data owners and domain experts, and are responsible for data quality and accuracy. This is similar to how microservices turn a set of technical capabilities into a product that can be consumed by other microservices. Implementing a data mesh on AWS is made simple by using managed and serverless services such as AWS Glue, Lake Formation, Athena, and Redshift Spectrum to provide a wellunderstood, performant, scalable, and cost-effective solution to integrate, prepare, and serve data.

One customer who used this data mesh pattern is JPMorgan Chase. For more information, see How JPMorgan Chase built a data mesh architecture to drive significant value to enhance their enterprise data platform.

Lake Formation offers the ability to enforce data governance within each data domain and across domains to ensure data is easily discoverable and secure, and lineage is tracked and access can be audited. The Lake House Architecture provides an ideal foundation to support a data mesh, and provides a design pattern to ramp up delivery of producer domains within an organization. Each domain has autonomy to choose their own tech stack, but is governed by a federated security model that can be administered centrally, providing best practices for security and compliance, while allowing high agility within the domain.

About the Authors

Nivas Shankar is a Principal Data Architect at Amazon Web Services. He helps and works closely with enterprise customers building data lakes and analytical applications on the AWS platform. He holds a master's degree in physics and is highly passionate about theoretical physics concepts.

Roy Hasson is a Principal Product Manager for AWS Lake Formation and AWS Glue. He works with customers around the globe to translate business and technical requirements into products that enable customers to improve how they manage, secure and access data.

Zach Mitchell is a Sr. Big Data Architect. He works within the product team to enhance understanding between product engineers and their customers while guiding customers through their journey to develop data lakes and other data solutions on AWS analytics services.

Ian Meyers is a Sr. Principal Product Manager for AWS Database Services. He works with many of AWS largest customers on emerging technology needs, and leads several data and analytics initiatives within AWS including support for Data Mesh.

The AWS Data Lake Team members are Chanu Damarla, Sanjay Srivastava, Natacha Maheshe, Roy Ben-Alta, Amandeep Khurana, Jason Berkowitz, David Tucker, and Taz Sayed.

Comments

T **The craft house**

6 months ago

"Lake Formation permissions are granted in the central account to producer role personas (such as the data engineer role) to manage schema changes and perform data transformations (alter, delete, update) on the central Data Catalog."

Observation by "The craft house" -> At consumer side also above mechanism needs to be maintained. Any changes on central Data catalog should be reflected at consumer side local Data catalog (specific to data domain) in automate fashion.

o o Reply • Share ›

KeithS

2 years ago

Hello guys (Roy Hasson, Roy Ben-Alta, Ian Meyer) from Keith.
Awesome work guys! This architecture looks scalable, highly flexible, and very promising for addressing a future project of mine. Thanks for putting this together!
cheers,
Keith

o o Reply • Share ›