Cloud Native Data Virtualization on AWS

Data Virtualization Guide

September 2020

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

This whitepaper discusses AWS practices, services, and benefits obtained from adopting a cloud-native data virtualization solution for an enterprise customer. The audience for this white paper includes IT leaders, enterprise data architects, and agile data practitioners seeking innovative ways to enable new business opportunities as part of a customer's journey to be a data driven organization. The typical customer profile is an enterprise customer with a heterogenous data landscape, which can include applications such as relational database management systems (RDBMS), Hadoop, data warehousing applications, and SAP, along with legacy enterprise applications like mainframes which have limited integration points.

Introduction

Data virtualization helps IT leaders and agile data practitioners unlock access to information in ways that drive timely insights into business performance and decision making. These are key factors to enable business agility and high performance in a data driven organization. Cloud-native data virtualization takes this a step farther by using AWS managed services and serverless components to implement a cloud-native data virtualization layer. Managing the infrastructure for a data virtualization solution hosted on-premise or in a cloud-based, self-managed environment can be a significant overhead for customers. AWS Managed Services enable organizations to quickly and easily deploy cloud infrastructure, resulting in the simplification of tasks such as provisioning, monitoring, and managing the security, availability, and business continuity of applications hosted in multiple regions world-wide. Serverless applications enable organizations to focus on their business requirements instead of on managing and operating servers. Enterprises can adopt AWS native features available in managed services such as Amazon Redshift and serverless components like Amazon Athena and AWS Lambda to orchestrate a lightweight, cloud-native data virtualization layer that reduces time to market and results in quick realization of business value at a lower cost. Adopting a cloud-native data virtualization strategy provides avenues for an enterprise to be more agile, responsive, and proactive to the needs of the business, and paves the way for the organization to innovate faster and stay on top of industry trends and competition.

This whitepaper uses terms and definitions familiar to AWS practitioners, data architects, and analysts, but the concepts presented can serve as a guide for anyone seeking to create an enterprise data strategy for a data driven organization.

An enterprise data landscape is composed of a variety of data formats, repositories, platforms, and capabilities that are built for corresponding tasks. At a high level, the pillars of an Enterprise Data Architecture are Data Collection, Data Curation, Data Preparation, Data Provisioning, Data Consumption, Data Governance and Security, and Data Monitoring and Operations.

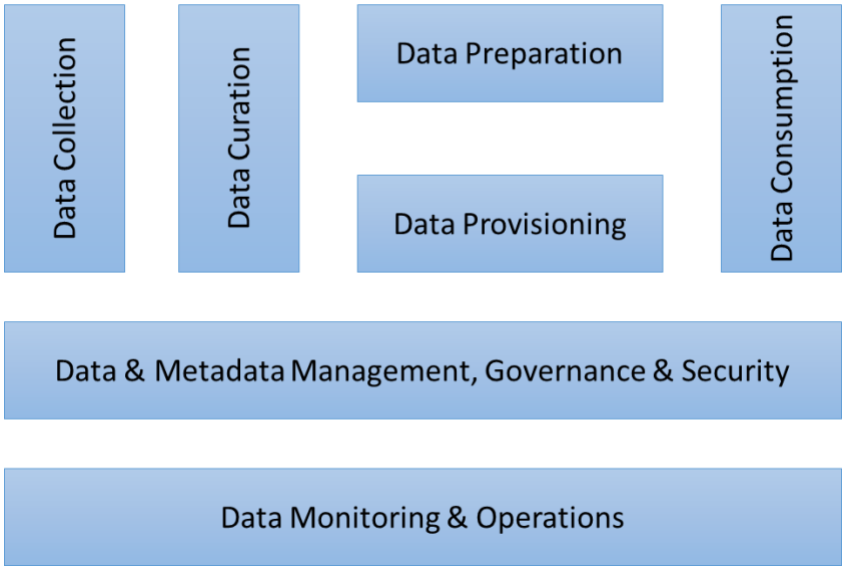

Figure 1 - Pillars of an enterprise data architecture

Over time, the data ecosystem developed many fit-for-purpose database systems. In the modern data landscape, data is no longer restricted to relational databases or line-of-business (LOB) systems. Data exists in the form of tens of file formats, hosted in hundreds of databases or servers, thousands of data messages exchanged per channel, and millions to billions of devices that stream data. With such a tremendous expansion in the data origination space, the variety of systems and platforms that process the data to enrich and optimize it, as well as the channels of consumption, are much broader. Figure 2 depicts an overview of the major components and platforms that make up a modern data landscape, where data may originate in one form and may pass through one or more of the components depicted in the diagram before ending its journey in the form of various consumption patterns. There are multiple means of storing and managing the data, and multiple channels for consuming the data.

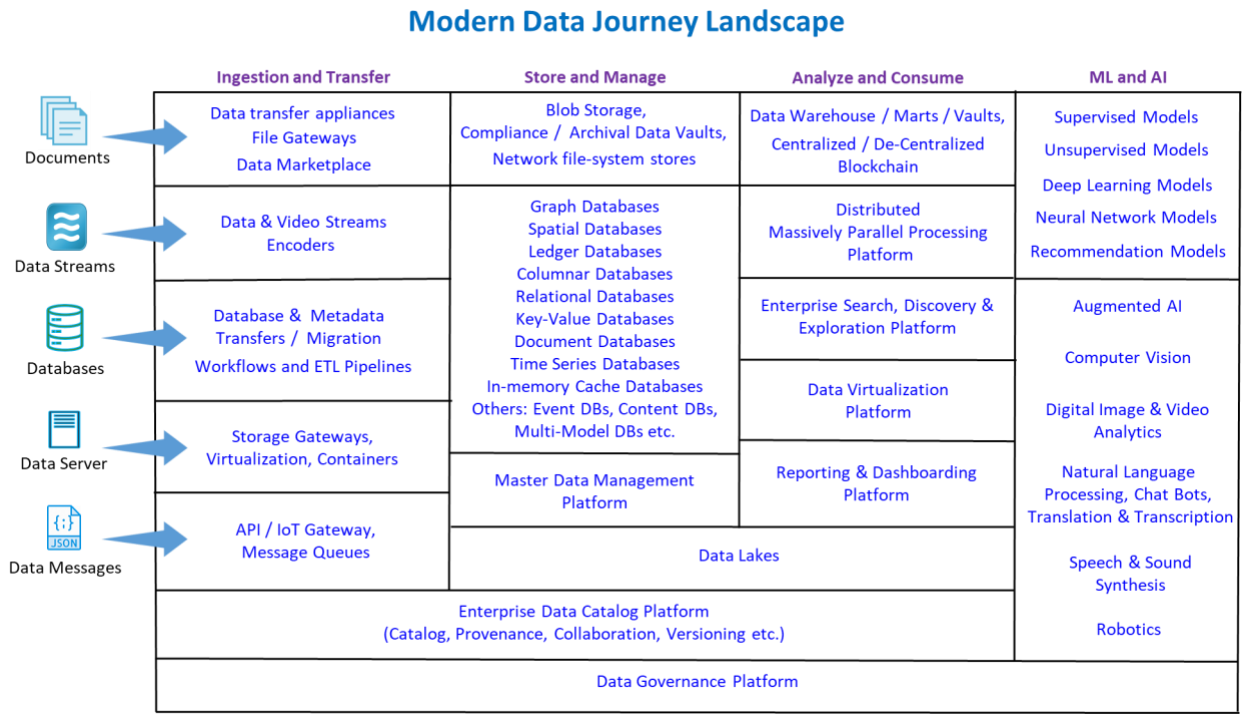

Figure 2 - Components and platforms of a modern data journey landscape

The traditional methodology of data integration involves several data hops from the data source, in which the data is physically transported, transformed, repeatedly replicated, and fine-tuned or optimized for the method of consumption. Large and complex datasets can be in tens of data formats, and terabytes to petabytes in scale. The traditional approach of transporting, transforming, and warehousing the entirety of the data in a one-size or one-type-fits-all data repository is time, resource, and cost-intensive. The ever-expanding volume, variety, and velocity of data poses some unique challenges.

Data Integration Challenges

There are numerous challenges associated with a typical data integration exercise that result in additional costs and overhead that impact business value and time to market.

- Data is typically hosted in a variety of fit-for-purpose data sources that may be located on-premise as well as in the cloud. When an end user wants to integrate data dynamically from different siloed source systems, the user often has to manually access these systems, and integrate the data locally using various data analysis or exploration tools.

- IT teams are engaged to operationalize this integration. This is often a point-to-point data integration exercise which is neither scalable nor cost effective.
- The traditional approach involves creating physical copies of data along with transformations. Consumption systems are tied to the last hop or copy of the data. In order to ensure that the data consumed is not stale, especially in cases where data needs to be made available in near real-time, a continuous or frequent data refresh cycle across all hops of data is required. This is resource intensive for large datasets, and often not scalable. Change Data Capture (CDC) is often the most complex and time-consuming process, and implementing CDC across all hops of data increases the cost and effort of implementation.
- With data getting replicated all over the data landscape, data governance becomes harder to manage, and leads to risks related to missed governance.
- With ever-evolving data integration needs, it takes too long to get answers, which causes a direct and significant impact to business. The result can be lost business opportunities and loss of revenue.

In summary, a data integration approach where data must be replicated, physically transported from one zone to another, and continuously transformed from one form to another is no longer a sustainable, scalable, or efficient approach.

Business Case

A data lake hosts data on a variety of repositories supported on the AWS Cloud, using services like Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (RDS), Amazon DynamoDB, Amazon Neptune, Amazon Quantum Ledger Database (Amazon QLDB), Amazon DocumentDB, Amazon Redshift, and other 3rd party databases or repositories hosted on Amazon Elastic Compute Cloud (Amazon EC2). All of these data sources host and serve many varieties, volumes, and velocities of data. With such a wide and heterogenous data landscape, users need the ability to integrate data from various sources, enrich and curate it using machine learning (ML) and artificial intelligence (AI) driven models, and access it from a unified interface that acts as a gateway to the enterprise data universe. AWS services like Amazon Redshift and Amazon Athena provide query federation features that enable data exploration across heterogeneous sources through a centralized interface. Amazon Athena also supports processing data using ML and AI services like Amazon SageMaker directly in the query language. This enables building a cloud-native data integration capability to

resolve a number of data integration challenges, and significantly reduces the time it takes to derive business insights.

Before diving deep into the technology, let's look at how these data integration challenges are addressed from a data architecture viewpoint.

Data Virtualization Architecture

Modern data architectures address the previously mentioned data integration challenges with a data architecture practice known as data virtualization. Figure 3 depicts a high-level data virtualization reference architecture emphasizing how and where data virtualization can fit into the overall data journey, depending on different data use cases.

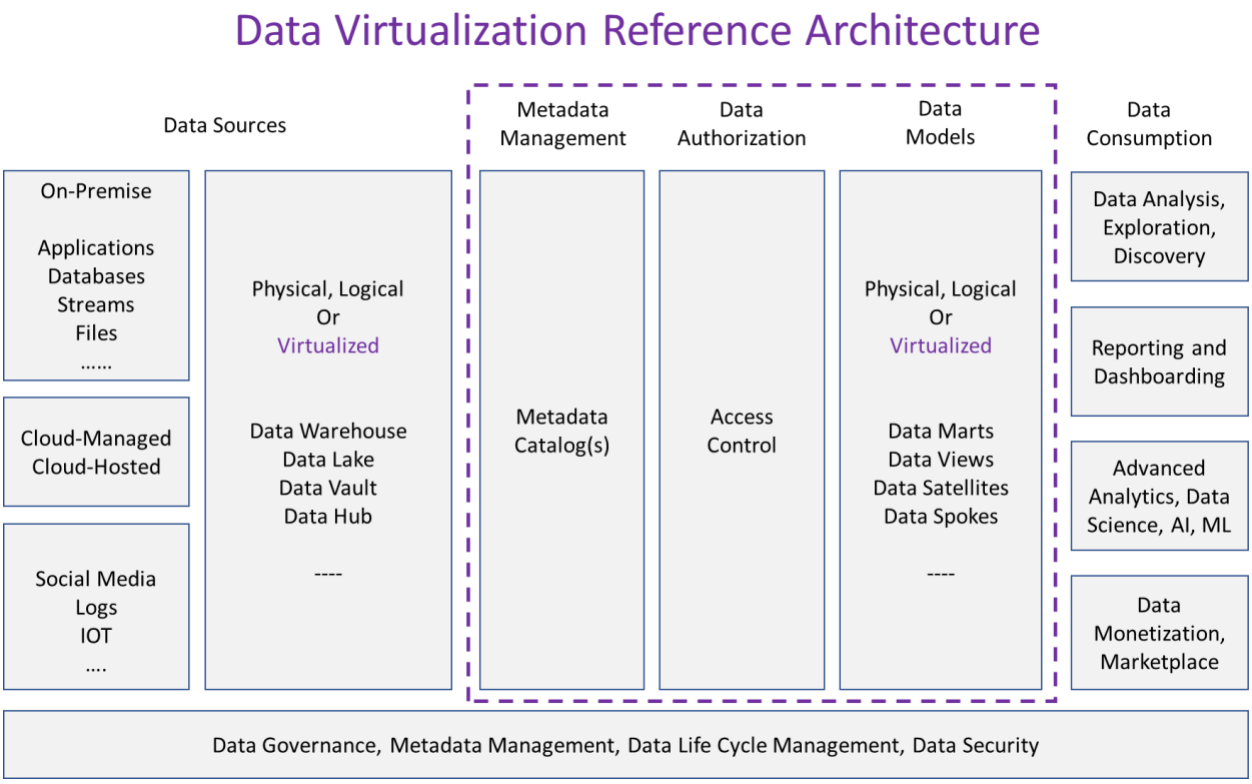

Figure 2 - Data virtualization reference architecture

The critical capabilities of data virtualization are as listed next.

- **Data abstraction:** An end user needs to be presented with a uniform and abstracted data layer, so that the data sources can be accessed in a uniform manner without accessing each siloed data source separately. From an end user experience, this is similar to accessing a single local data source with the ability to create a virtual model by combining data objects from various data sources.
- **Real-time and On-Demand data access:** The data accessed using the abstracted layer should not be stale, should be capable of supporting data at rest or in motion arriving through streams, and should be accessible on demand without the need to engage an IT team for point-to-point or case-specific data integration.
- **Centralized metadata, security and governance:** As the data is presented to the user via an abstracted layer which provides a unified view, the metadata, security and governance should be integrated with the abstracted layer as well.
- **No data replication or relocation:** Any physical data replication or relocation involving transformation will not be required, which avoids typical data integration challenges.

As data virtualization reduces the need to physically replicate or relocate data, this results in significant cost and resource savings, as outlined in the bulleted list.

- Frequently accessed data (also known as hot data) is stored in purpose-fit repositories like Amazon Redshift and Amazon Elasticsearch Service, and data that is accessed infrequently (also known as cold data) is stored in general purpose storage like Amazon S3 or Amazon RDS. In the absence of data virtualization, cold data is often physically replicated and relocated along with hot data for data integration and exploration purposes. As data virtualization enables an integrated view of data originating from multiple data sources, there is a reduced or limited need to physically replicate or relocate data. Because of this, the category of batch jobs that are primarily targeted for replicating or relocating data can be mostly eliminated. This results in significant savings in terms of development, maintenance, infrastructure, licensing and operational costs.
- Because the data model and the underlying data is not replicated either directly or indirectly, supporting data architecture components like the metadata catalog, user roles and policies, data format conversions, and data archival strategies are easier to implement. This approach is less resource intensive and more manageable to operate due to a comparatively smaller data footprint.

As a result of these, speed to market is significantly enhanced, resulting in faster business value realization. Considering this in the context of the cloud, there are two options to implement data virtualization on AWS.

Custom Data Virtualization Solutions

The first option involves developing a build-your-own or custom data virtualization solution on AWS, which can be challenging and resource-intensive to maintain. Developing a custom data virtualization solution for your organization requires developing some basic components:

- A common user interface (UI) for end-user interaction.
- Building a metadata catalog and registering all data sources with it, or integrating UI with an existing corporate metadata catalog.
- Custom built interfaces like APIs that accept end user requests and route them to a query optimization engine.
- A full-fledged query optimization engine that accepts queries from the UI and translates them into data source specific queries, executes the queries, and collects the output.
- A caching layer that caches the data and returns it back to the calling interface.
- A security layer that talks to the security repository and checks for granular level permissions.

Commercial Data Virtualization Solutions

The second option is to utilize a commercial data virtualization vendor solution. Some examples of commercial vendors in the data virtualization space are Tibco Data Virtualization, Denodo, Actifio and Data Virtuality's Logical Data Warehouse. A generic high-level architecture of a commercial data virtualization solution option is shown in Figure 4.

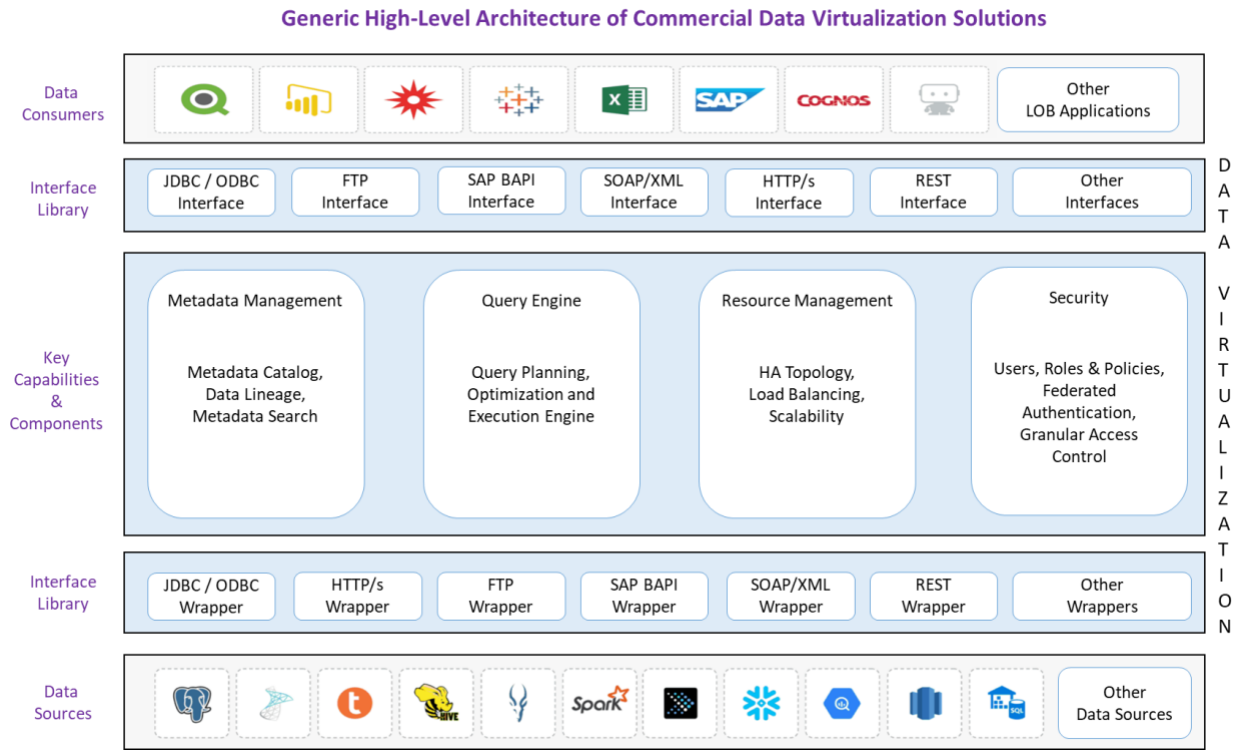

Figure 3 - Generic high-level architecture of a commercial data virtualization solution

A third option involving an orchestrated or managed cloud-native data virtualization layer would be of value for clients who can't pursue the other two options due to cost and time constraints. The top three factors that may lead to adoption or preference of a cloud-native data virtualization solution are listed next:

1. Architectures that don't have a large-scale federated data footprint on-premise as well as on the cloud may find that the complexity and cost of implementing a commercial data virtualization solution outweighs the benefits. Many commercial data virtualization solutions offer Pay As You Go, Bring Your Own License and serverless metered pricing options, as per Gartner's 2019 Magic Quadrant of data integration tools. This report notes that many challengers and leaders saw a steady increase in their subscription or license costs, and observed scalability challenges as well.
2. The following process is a costly, resource-intensive, time-consuming effort:
    a. Integrating commercial data virtualization solutions with corporate metadata catalogs.

   b. Binding these commercial data virtualization solutions with federated authentication.
   c. Calibrating with corporate governance, security standards, and policies.
   d. Setting up a production installation environment.
   e. Rolling out to the intended user base.

3. Organizations that are in the process of migrating or modernizing their data architectures with cloud-based services may not have the appropriate maturity and/or readiness to efficiently absorb and integrate a data virtualization solution at scale.

AWS Cloud Native Data Virtualization Solution

Considering the previously mentioned points, many customers may benefit from a light-weight and cloud-native data virtualization layer that is neither too complex to implement nor too subtle to be considered as a full-blown data virtualization solution. AWS has native services and features which can map to many of the capabilities offered by commercial data virtualization solutions. These native features may not be an exact mapping or replacement for these commercial solutions, but they have the potential to address many of the core capabilities desired in a data virtualization solution. AWS features like the newly introduced Federated Query and User Defined Function, available in services like Amazon Redshift and Amazon Athena, coupled with AWS services like AWS Lake Formation, Amazon Athena Query Federation SDK, AWS Glue and AWS Lambda pave the way for orchestrating a light-weight cloud-native data virtualization solution. Figure 5 shows a mapping of how these AWS services fit into the overall data virtualization solution landscape when compared to commercial data virtualization offerings.

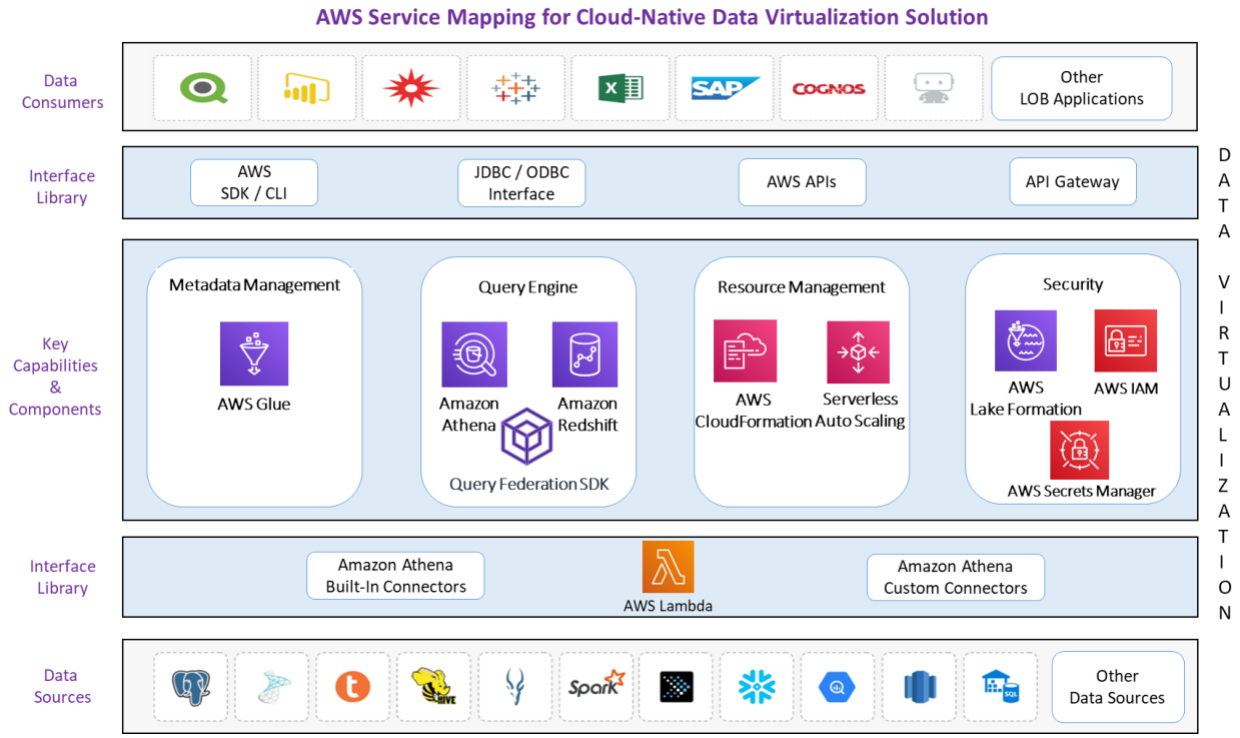

Figure 5 - AWS service mapping for cloud-native data virtualization solutions

AWS Services for Cloud-Native Data Virtualization

Let's consider the data landscape for a typical large enterprise customer on AWS that has heterogenous forms of data that is generated, processed and consumed in a variety of ways. Human generated data, such as transactional data, may be generated directly by applications or imported from third party sources into an Amazon Relational Database Service (Amazon RDS) database instances. Any underlying metadata or configuration data may be held in Amazon DynamoDB. Data imported into the data lake may land on Amazon S3 and eventually make its way to more appropriate databases like Amazon DocumentDB, Amazon QLDB, Amazon Neptune, Amazon Timestream, or other database systems and services supported on AWS. A part of this data might make its way into systems like Apache HBase on Amazon EMR, Amazon Redshift and Amazon SageMaker for deriving business insights. Machine learning models are a critical capability to augment data in modern data processing architectures, which means data can be analyzed using Amazon Sagemaker models and inference engines before it is served to the end user. All the underlying data here represents the actual

business data that is housed in various fit-for-purpose data, analytics, and machine learning service components.

Machine generated data consists of logs, metrics, configuration inventory of virtual machines (Amazon EC2), data volumes, and so on. Depending on the sensitivity of data, a significant portion of data may reside either permanently or semi-permanently on premise instead of being migrated to the cloud. Another layer, AWS Data Exchange, can serve as a major channel for providing external or reference datasets to support data analysis and exploration for end-users. From a data integration perspective, different user personas may use different datasets which may be hosted in different data repositories in the cloud, on-premise, and with external data providers.

As part of a customer's strategy to be an agile, data-driven organization, one of the guidelines could be based on the adoption of a *cloud-native first* approach. As part of this strategy, an organization may be interested in implementing a cloud-native data virtualization approach. This approach may not fully conform to all the data virtualization principles and reference architectures outlined earlier, but it should honor many of the core capabilities of a data virtualization solution offering. Let's now dive deep into the AWS data offerings to understand how these components can be used to create a logical, cloud-native data virtualization landscape. Figure 6 provides a view of this architecture, enabled using AWS native federated query features and user-defined functions available in Amazon Athena and Amazon Redshift. As identified earlier, one of the key benefits of this approach is the ability for end users to explore and query data from a variety of data sources and tap into native integration capabilities using AWS services like Amazon SageMaker and Amazon QuickSight.

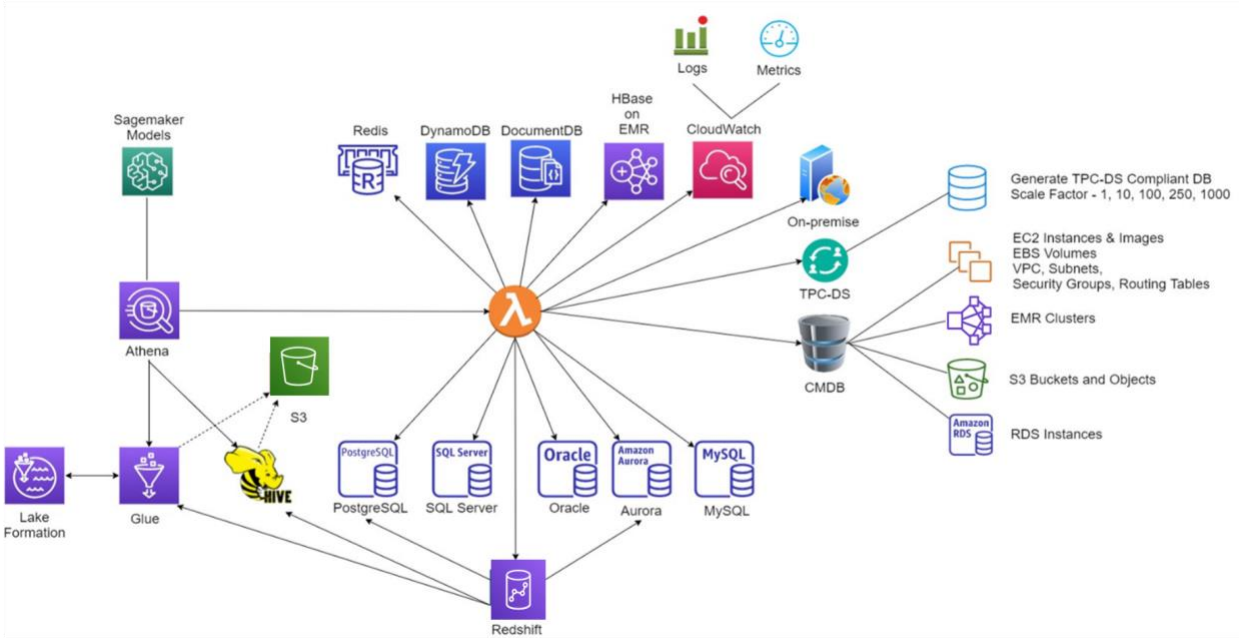

Figure 6 - Cloud-native data virtualization with federated queries and user-defined functions

Use the following points to interpret Figure 6:

1. Amazon Athena and Amazon Redshift provide Java Database Connectivity (JDBC) drivers which can be used with notebooks and Integrated Development Environments (IDEs) like Jupyter, RStudio, SQL Workbench, Aginity, Jetbrains DataGrip and others for the purposes of data exploration. Amazon Athena and Amazon Redshift also support JDBC connections, which can be used programmatically to access data from external data sources.
2. Amazon Athena and Amazon Redshift have native integration with the AWS Glue metadata catalog as well as the Hive Metastore to access data from Amazon S3. AWS Lake Formation acts as an authorization layer on the top of the AWS Glue data catalog, and permissions administered by AWS Lake Formation are honored by both these services. Figure 7 illustrates the built-in integration with the AWS Glue and Hive metastores in Amazon Athena.

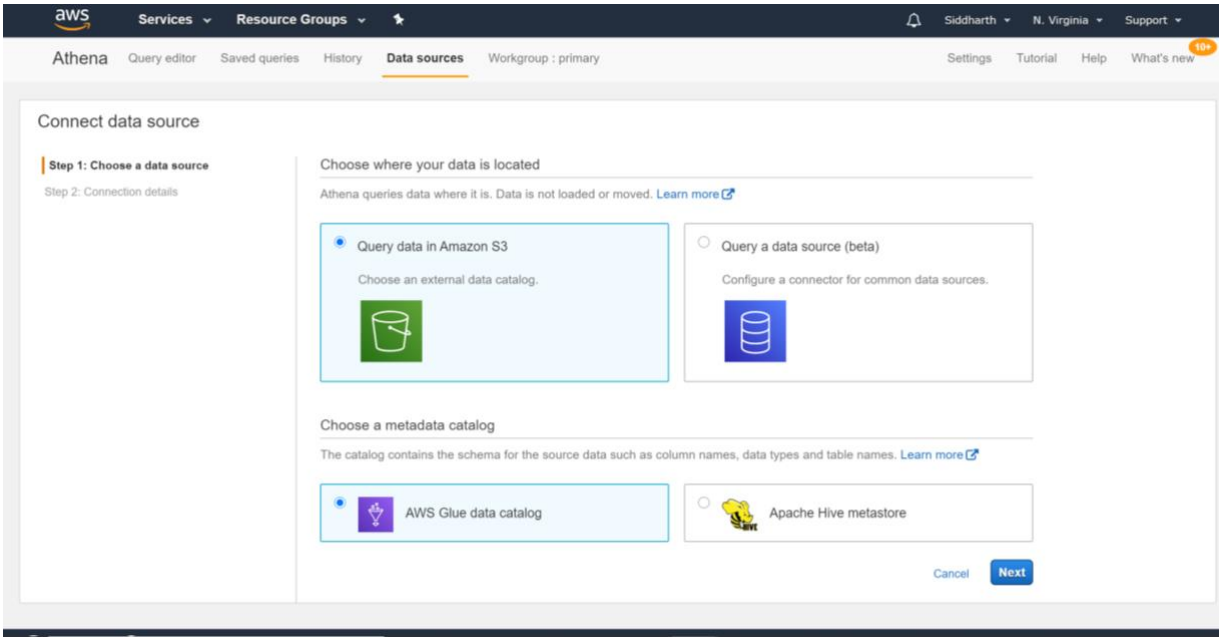

Figure 7 - Built-in integration with AWS Glue and Hive metastores in Amazon Athena

3. Amazon Redshift introduced the federated query feature in December of 2019. This feature provides a mechanism to query data stored on Amazon S3 and register the data with the AWS Glue data catalog. Data stored locally in Amazon Redshift, data stored in Amazon RDS PostgreSQL Instance, and data stored in Aurora with PostgreSQL compatibility can also be queried.

   a. **For data stored in Amazon S3:** You can register the intended external schemas on Amazon Redshift. Once the required privileges are granted in AWS Lake Formation, these schemas and objects are accessible in Amazon Redshift. Figure 8 shows an example of one such schema.

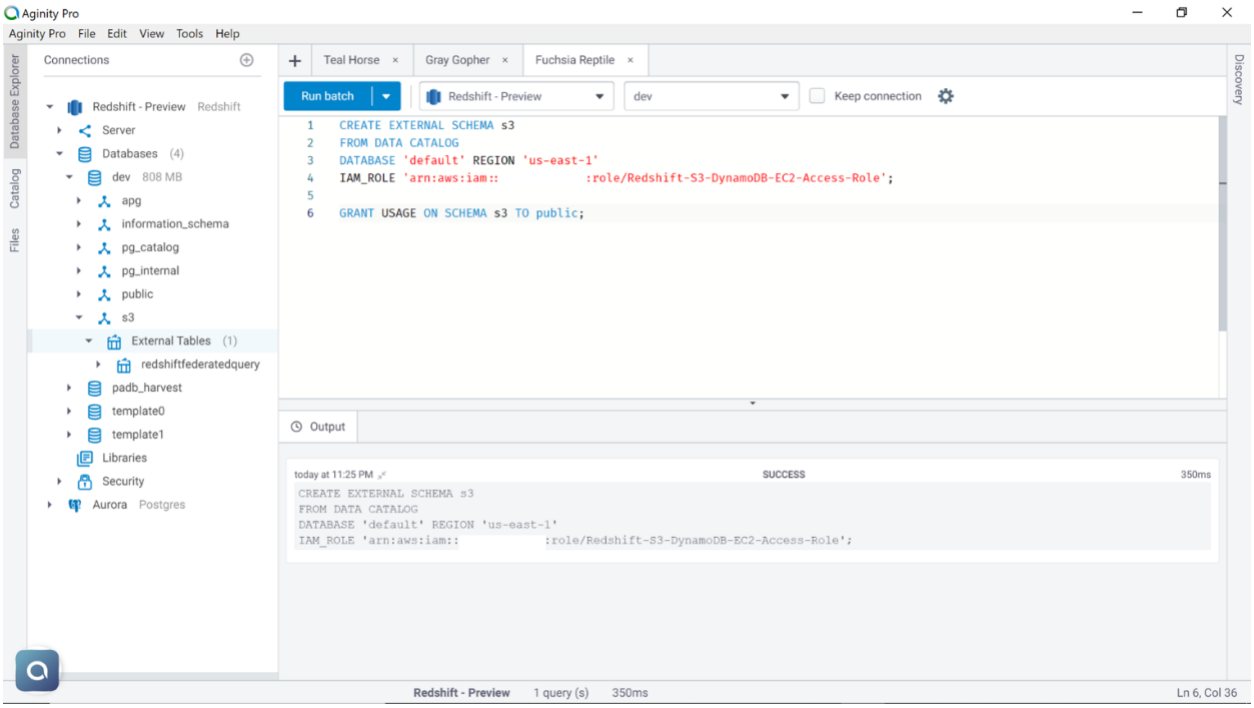

Figure 8 - A schema registered on Amazon Redshift

b. **For data stored in Amazon RDS PostgreSQL / Aurora:** You can register the database credential as a secret in AWS Secrets Manager and provide enough access to the role used by Amazon Redshift to access the secret as well as the RDS PostgreSQL / Aurora instance. Figure 9 shows an example of this, where an external schema named `apg` is created using the secret stored in AWS Secrets Manager.

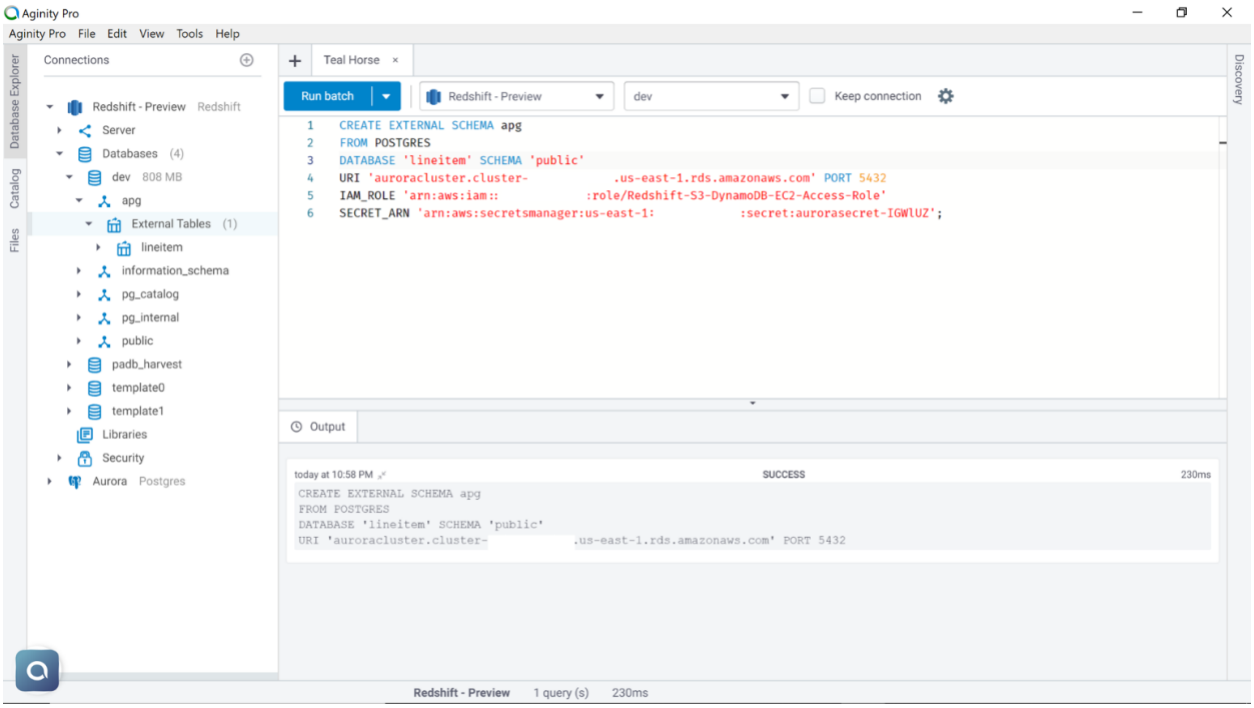

Figure 9 - External schema created in AWS Secrets Manager

c. Once the schemas are created, you can create a view (which can be seen as a virtualized object from an architecture perspective) by joining datasets in the underlying schemas. The datasets will point to actual data repositories like Amazon S3, Amazon Redshift local storage and Amazon RDS (Postgres / Aurora). Figure 10 shows an example of this.

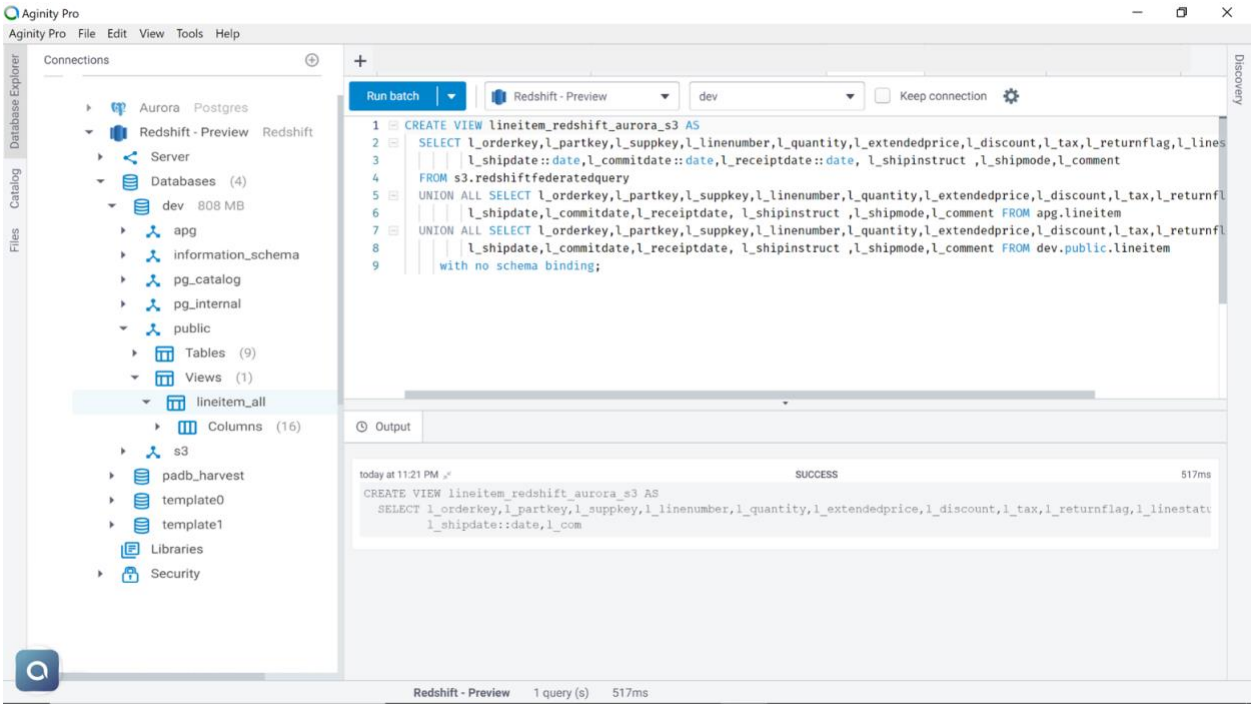

Figure 10 - View created by joining datasets in the underlying schemas

d. Once the view is created, end users can access and query the data from these virtualized data objects, and results are returned based on the assigned privileges for that end user. Figure 11 shows an example of the result of one such federated query.

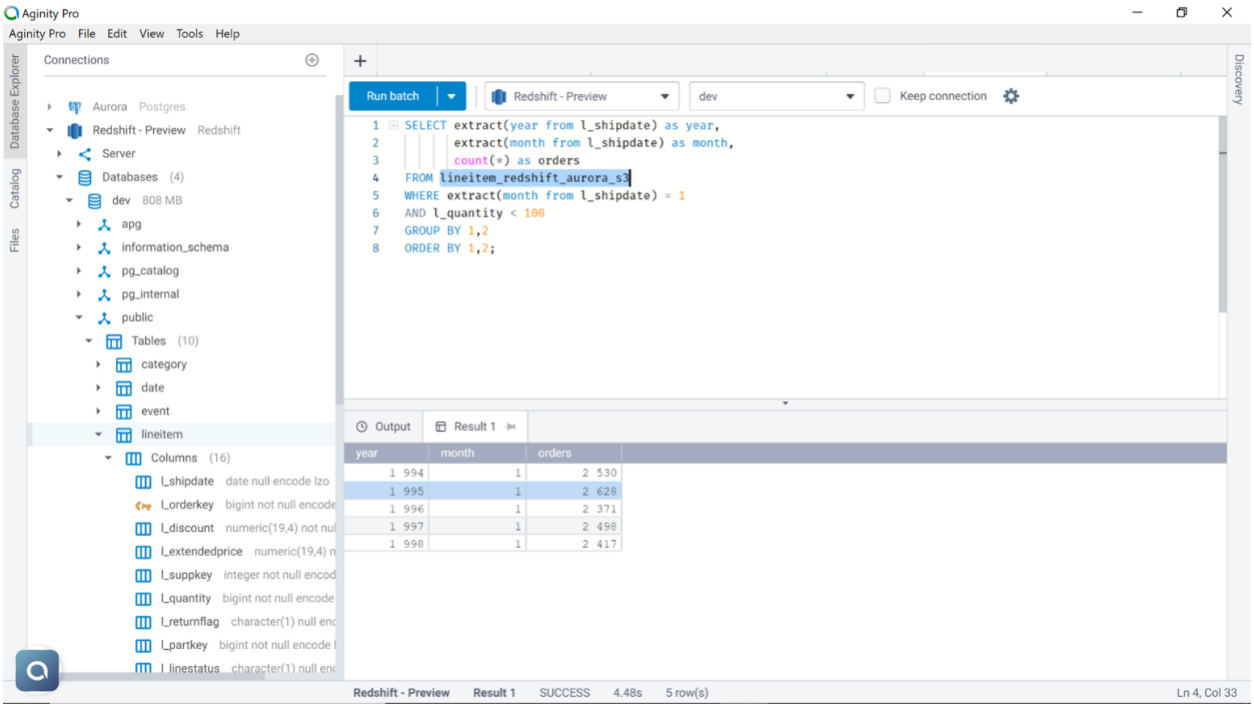

Figure 11 - Federated query result

Amazon Athena introduced federated query in November of 2019. Amazon Athena supports many data sources, as shown in Figure 12. In the AWS cloud-native data virtualization reference architecture depicted in Figure 6, you can see how these data sources can be virtualized using AWS Lambda functions. There are connectors[1] available for many of these data sources today, and Amazon Athena also allows the creation of a custom connector for any specific type of data source. These connectors can be deployed via the Serverless Application Repository[2] as well.

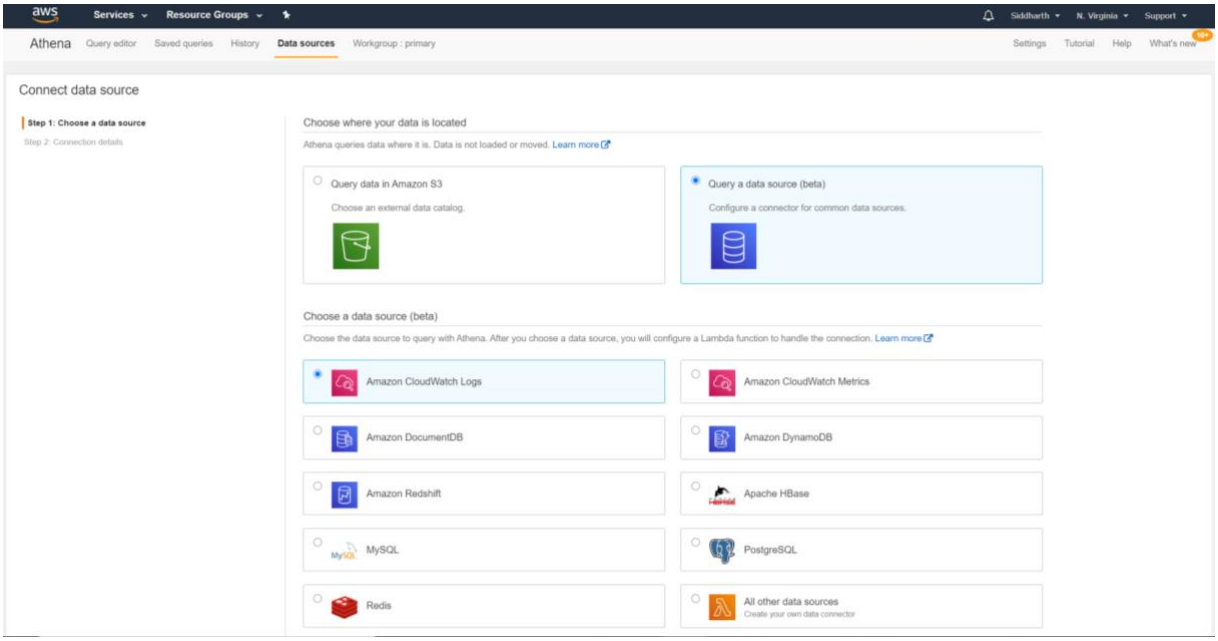

Figure 12 - Data sources supported in Amazon Athena

4. Once you select a data source, create a new AWS Lambda function or re-use an existing once, depending on the nature of the data source or connector. Figure 13 illustrates this step.

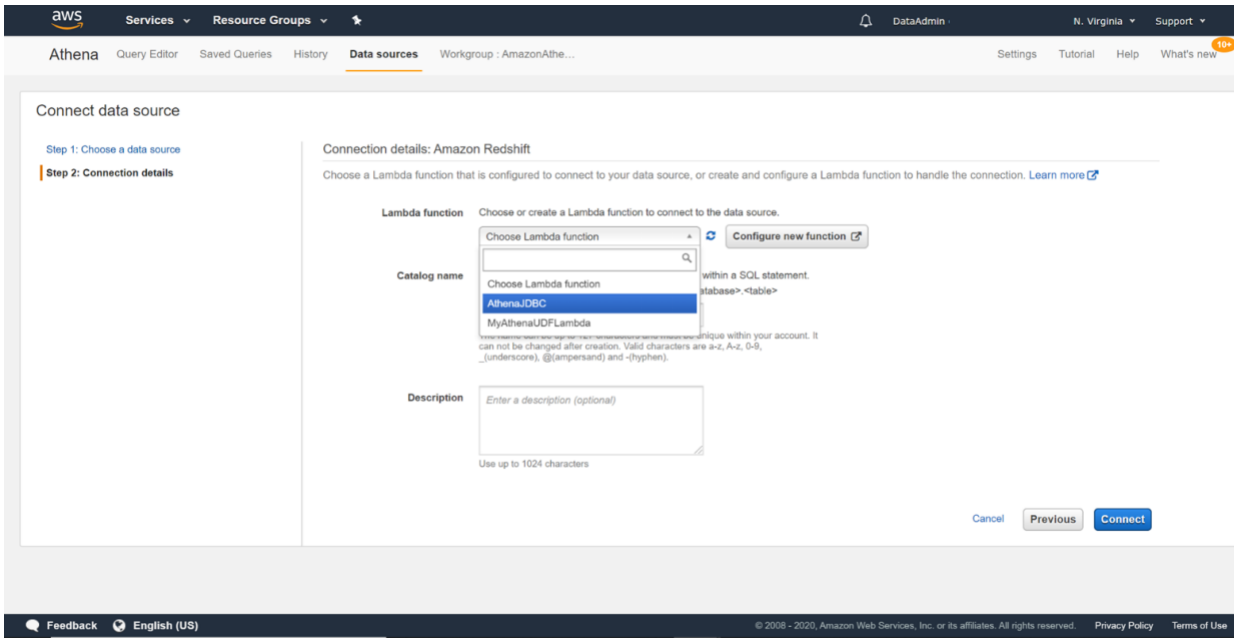

Figure 13 - Create a new AWS Lambda function or configure an existing one

5. After you configure the AWS Lambda-based connector with the required data source endpoints and credentials, it's listed in the data sources for use with Amazon Athena. Figure 14 shows an example of registered data sources.

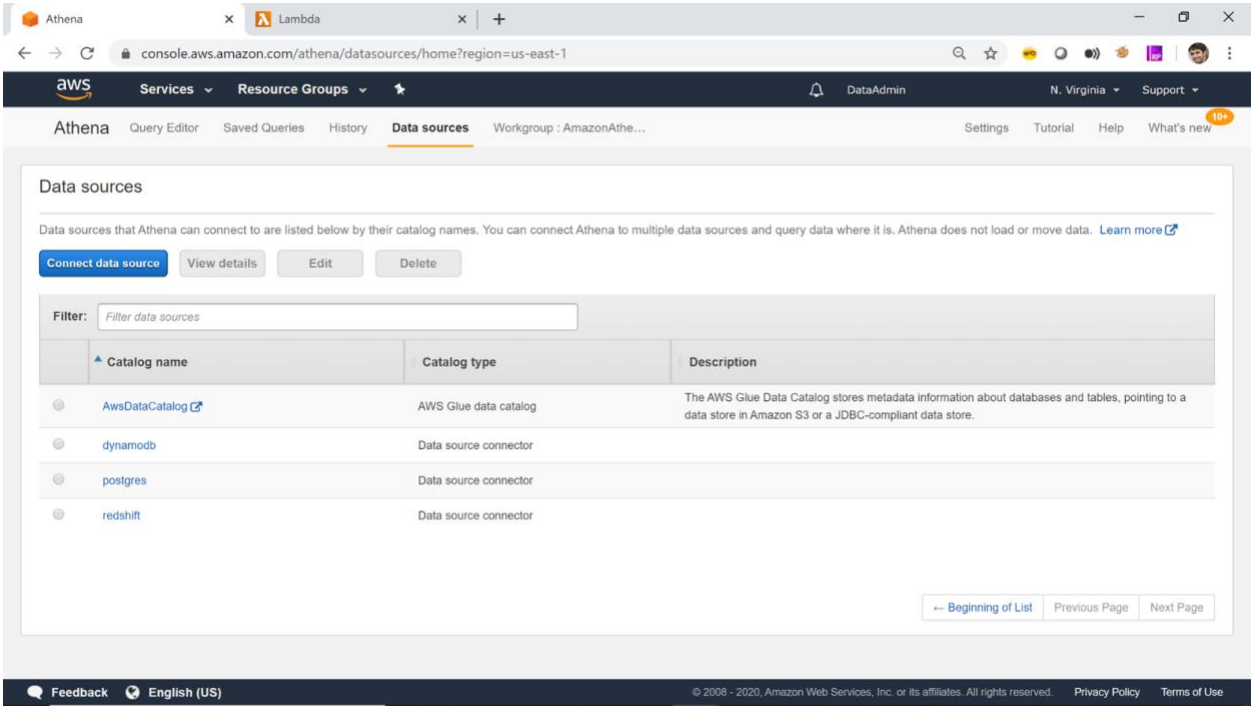

Figure 14 - Registered data sources

6. Once registered, these data sources can be queried directly from Amazon Athena, as seen in figure 15.

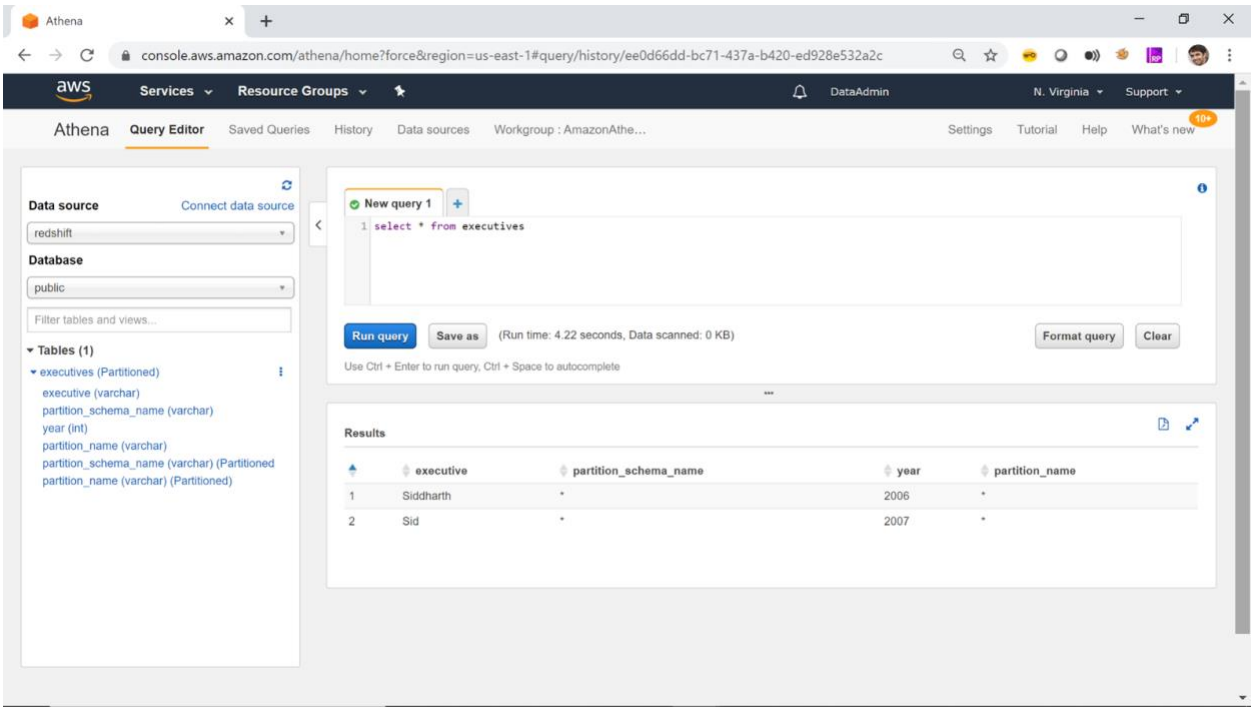

Figure 15 - Query data sources from Amazon Athena

7. Typically, users access Athena using Integrated Development Environments (IDEs) of their choice. These registered data sources act as a virtual layer for end users who need not be aware of the details around where these sources are hosted or any complexities involved in integrating with them. End users can execute a federated query that joins all these data sources. Depending on their privileges in the respective data sources, end users can also query and explore the data originating from the source with Athena acting as the bridge. Figure 16 shows an example query that joins data from Amazon S3, Amazon Redshift, Amazon DynamoDB, and Amazon RDS Postgres, where selected attributes from all these data sources are returned based on a specific criterion. In a sample performance test, this query performed complex join operations on four different repositories using around 16k records, and returned results in about 11 seconds. These query results are not optimal but they provide an initial estimate. Architects and performance engineers can fine-tune their query performance based on the data volumes and performance needs of their use cases.

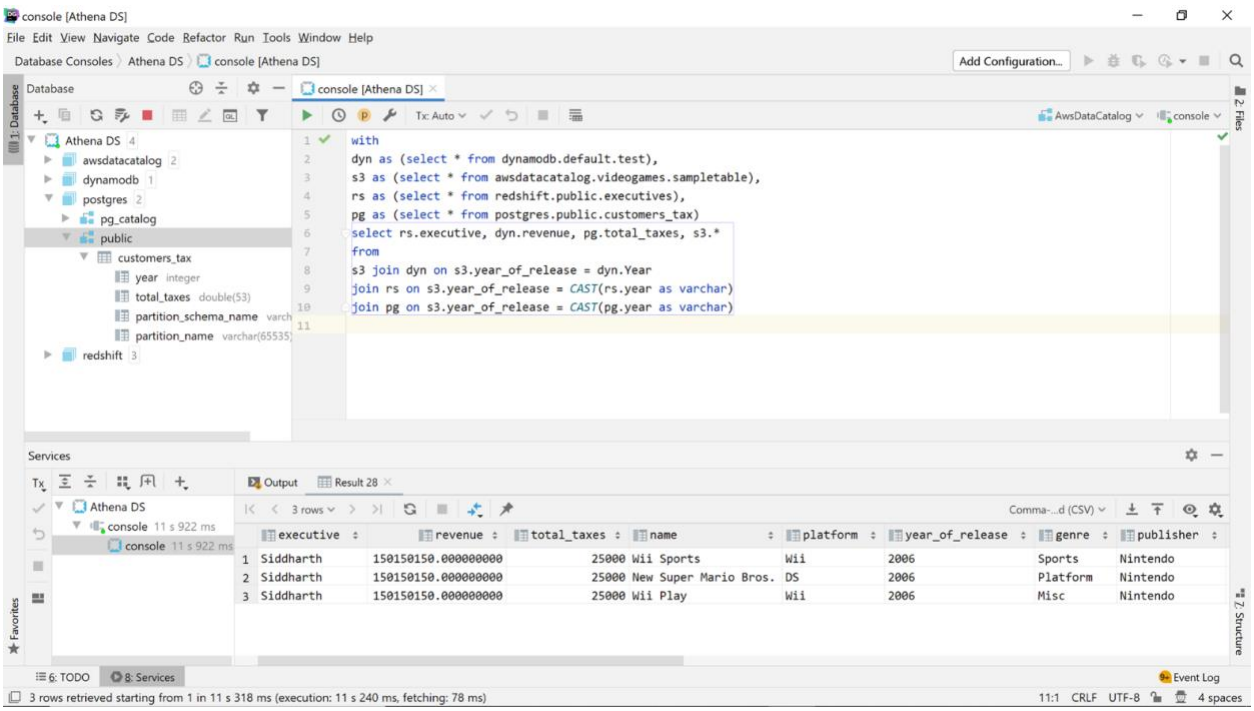

Figure 16 - Federated query results

8. Enterprises can use this feature in Amazon Athena to register data sources such as mainframe, SAP, and others typically found in a large enterprise environment. This enables users to explore and query enterprise data assets such as Virtual Storage Access Method (VSAM), DB2 for z/OS, Information Management Systems (IMS) DB, Adabas and other legacy data stores that are hard to tap into. These types of data stores often remain unexplored due to their complexity and limited integration points. It is worth noting that Lambda has the ability to support frameworks like BlueAge that can execute Cobol functions using AWS Lambda.

Machine learning is becoming a mainstream component in the way data is queried and analyzed to derive business insights. Amazon Athena supports invoking Amazon Sagemaker[3] machine learning models directly in the SQL Query by using user-defined functions.

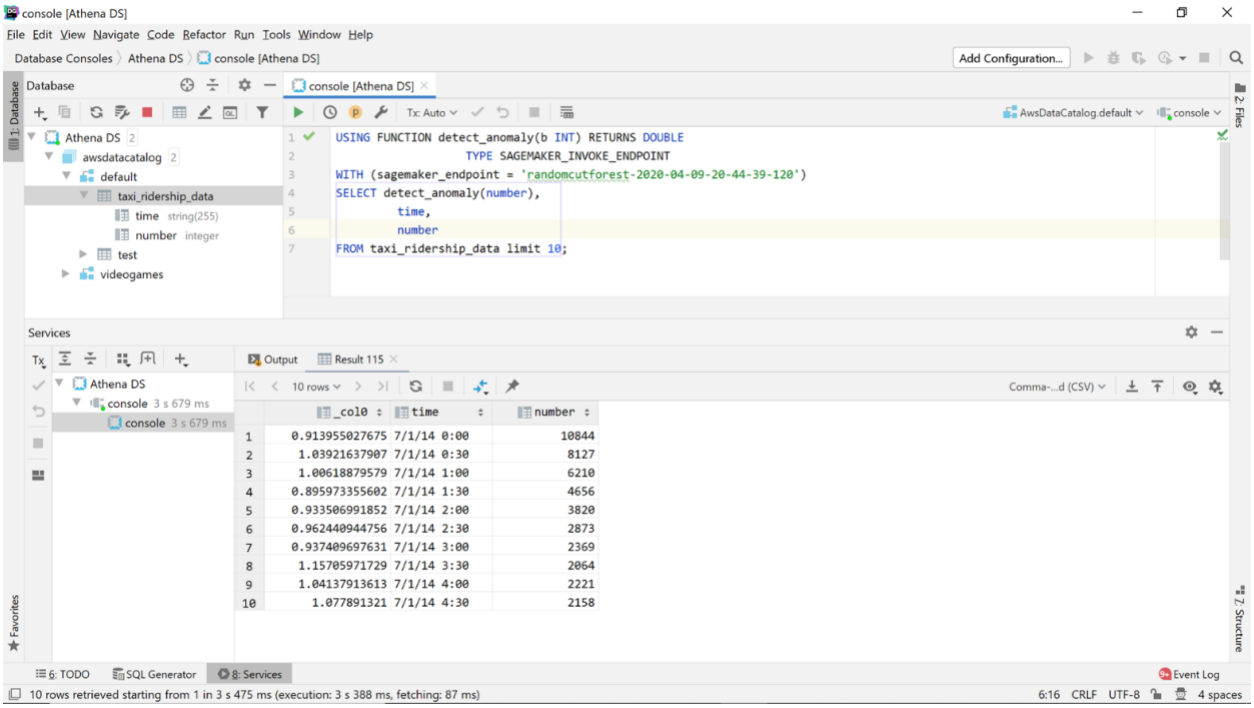

Figure 17 - Use user-defined functions to invoke Amazon Sagemaker machine learning models

Validation of Core Capabilities

Now that you understand how these services are integrated and how they provide the foundation for a virtualized solution, let's do a final check on whether this approach is compliant with the critical data virtualization capabilities identified earlier. Table 1 summarizes how this approach addresses each of the core capabilities of a data virtualization solution.

Table 1 - Core data virtualization capability table

Core data virtualization capability	Is the core capability addressed in the cloud-native approach?	Solution approach
Does the approach eliminate or reduce the need to physically replicate or relocate data for query and exploration?	Yes	Almost no data replication or relocation is needed. Data can be queried without physically copying data to a central repository.

Core data virtualization capability	Is the core capability addressed in the cloud-native approach?	Solution approach
Does the approach reduce dependency on IT teams?	Yes	Only a one-time IT setup is required. There is no IT dependency for each point-to-point Integration aspect.
Does the approach provide a single point of access to query and explore the data?	Yes	End users can access the data from a single location without managing connection credentials separately.
Does the approach provide live and on-demand data?	Yes	As the data is queried directly from the data source, the queries always deliver live results.
Is the approach governed by a metadata catalog and permission framework?	Yes	The data access model is governed by the Amazon Glue metadata catalog and the AWS Lake Formation permissions and security model.
Is the approach extensible to support connections to cloud-native and on-premise data sources?	Yes	The approach makes use of Amazon Athena Query Federation SDK, an extensible framework that supports developing custom connectors which can connect to cloud-native and custom data sources.

These questions ascertain the conclusion that by using federated query and user-defined functions, AWS features provided by services like Amazon Athena and Amazon Redshift can be used to implement a cloud-native virtualization solution.

Cost Savings

The next question that arises is the potential extent of cost savings. Accenture, a Global System Integrator (GSI) and an AWS Partner Network (APN) Premier Consulting Partner, had an engagement with an enterprise customer who wanted to evaluate their end user query performance, along with speed of data delivery and potential cost savings for a use case that involved querying multiple data sources. The team performed a high-level analysis to determine the savings involved with a cloud-native data virtualization solution in place. Table 2 depicts some of the data points gathered from the analysis. As part of this effort, a relatively small footprint of data from multiple

data sources was identified: approximately 100 tables, 80 stored procedures, and 50 views.

Table 2 – Cost savings data points

Data integration task	Level of effort without data virtualization in person days	Level of effort with cloud-native data virtualization in person days
Analysis and profiling of data source(s) to be integrated	10	10
Incremental data refresh analysis and optional snapshot setup for sources that do not support CDC	10	Not Applicable
Design and build of data model for the target data warehouse and/or data mart	5	Not Applicable
Data dictionary of source and destination data models	5	5
Metadata and lineage tracking	5	Not Applicable
Schema compatibility analysis	1	1
Infrastructure, tools, and technology stack setup for batch jobs	3	Not Applicable
Design and build of batch jobs	15	Not Applicable
Scheduler and data refresh jobs setup	3	Not Applicable
Execution log storage for auditing	3	Not Applicable
Monitoring of batch job execution logs and alerts setup	3	Not Applicable
Snapshots for point-in-time disaster recovery of target sources	5	Not Applicable
Maintenance of data mart / warehouse / batch jobs to keep source and target schema in sync	5	Not Applicable

Data integration task	Level of effort without data virtualization in person days	Level of effort with cloud-native data virtualization in person days
Identifying datasets fits for use with a data virtualization layer	Not Applicable	5
IT Overhead of integrating data sources in a cloud-native data virtualization layer for 100 Users	Not Applicable	5
Total	73	26

This calculation shows a potential of up to 60% savings. If you consider this to be an optimistic calculation, you can factor in aspects such as complexity in processing, setup costs, and any unforeseen overheads and still project cost savings ranging from 20% to 30%. From a performance standpoint, as most of the data integration (ETL) tasks are eliminated with data virtualization, the remaining tasks can be executed in parallel without much interdependency. This approach has the potential to deliver a 30% to 50% increase in speed of delivery once the data virtualization approach has been industrialized.

Conclusion

Cloud-native data virtualization can prove to be a compelling option for a significant number of enterprise customers in their journey to the cloud. Notably, it can serve as the bridge between developing your own custom data virtualization layers versus buying commercial data virtualization solutions. In addition, it has the potential to support the scale and growth needs of an enterprise data platform with ever-increasing needs to support a variety of heterogenous data sources.

In this article, we have outlined various data consumption and delivery challenges leading to the adoption of data virtualization solutions. We've made the case for using cloud native features in AWS to serve as the foundation for a cloud-native, light-weight data virtualization solution. There is ample scope for an enterprise to explore and adopt a cloud native approach toward data virtualization. We don't intend this to be a direct replacement for commercial data virtualization tools that may offer more features and enhanced support for complex data virtualization initiatives. However, depending on the use case and factors such as CAPEX and OPEX spend, an enterprise can fine-tune their data virtualization approach based on their overall data strategy, and implement a

cloud-native data virtualization solution that can result in faster insights into data, immediate cost savings, increased business agility, and competitive advantage.

Contributors

The following individuals and organizations contributed to this document:

- **Siddharth Mehta, Master Technology Architect, Accenture**
- **Gopal Wunnava, Principal Solution Architect, AWS**

Further Reading

For additional information, see:

- Gartner Magic Quadrant for Data Integration Tools
- Data Virtualization is the CDO's Best Friend
- AWS Managed Services FAQs
- Serverless

Document Revisions

Date	Description
September 2020	First publication

Notes

[1] https://docs.aws.amazon.com/athena/latest/ug/athena-prebuilt-data-connectors.html

[2] https://docs.aws.amazon.com/athena/latest/ug/connect-data-source-serverless-app-repo.html

[3] https://docs.aws.amazon.com/athena/latest/ug/querying-mlmodel.html