Paper 4634-2020

SAS® on an Amazon Web Services Data Lake: Enabling Access to Data Lakes on SAS

Dilip Rajan, Amazon Web Services

ABSTRACT

Modern business applications produce a large variety of data at different volumes and speeds. Traditional data warehouses were not designed to consume and process such large amounts of data. Due to their tightly coupled storage and compute nature, customers end up having to pay for data at rest even if that data is not frequently queried or used. Data lakes provide the ability to store data in raw format, which enables a wide variety of use cases such as real-time analytics, machine learning, batch processing from different sources such as log files, clickstream events, and so on. SAS® and its integration with the AWS data lake solution (Amazon Simple Storage Service, or Amazon S3) provides users with the flexibility to access data in disparate formats using AWS Glue. AWS Glue is a fully managed service that automates the time-consuming steps of data preparation by automatically discovering and profiling your data into a metadata catalog and transforming it into the target schemas/destinations. This paper provides an overview of data lake components within AWS and how they can be used with SAS for various use cases

INTRODUCTION

Data creation is expected to grow to 163 ZB by 2025 however, most companies analyze only 12% of their data. At the same time, customer's data is fragmented and they require single-purpose data silos to store and analyze structured and unstructured data. Structured data traditionally exists in relation databases while unstructured data is in the form of emails, images, video, instant messages and social media content. This causes the data to exist incompatible formats and is difficult to analyze together. In addition, customers also want to monetize their data but don't have enough expertise to build a Data Lake or deploy the physical infrastructure and tools required to get started. On-prem solutions and data lake environments don't provide customers with adequate data protection, resiliency or security controls. Above all this, it takes too long for customers to aggregate and derive value while preparing for data analytics

In this paper, we will review the various technologies within AWS that help customers build a data lake and services that can help derive value from their data lake. Readers would get an overview of the various services provided by AWS and their integration with SAS products reducing their time for aggregation and analytics.

AWS DATA LAKE – S3

AWS Data Lake in S3 (Simple Storage Service) is an object storage service that offers scalability, data availability, security and performance. S3 is designed for 99.99999999999% of durability and stores data for millions of applications for companies all around the world. With S3, a data lakes becomes a centralized repository that allows you to

migrate, store, manage all structured and unstructured data at unlimited scale and then gain insights through analytics and machine learning.

Following are the key benefits of building data lakes on AWS S3:

1. Cost Optimization – S3 provides more storage tiers including S3-intelligent tiering which automatically moves data between different storage tiers based on workflow and how data is being accessed. Glacier Deep Archive provides exceptionally cost-effective long-term data retention
2. Security, compliance and audit capabilities – With a combination AWS IAM, Glue and Lake Formation, S3 provides granular data and metadata access controls. S3 public block ensures that data can't be inadvertently be exposed to the public and it is the only service that lets you see every API access to data and management events. S3 also integrates with Macie which is a machine learning service used to identify and classify data into different categories if there is sensitive data.
3. Object Level Control – S3 lets you control data access management at the fine-grain level of individual objects. By setting editable S3 object tags, you can do almost anything – like tier objects to lower cost storage, replicate only the most valuable data using CRR or provide access controls down to the individual object level.

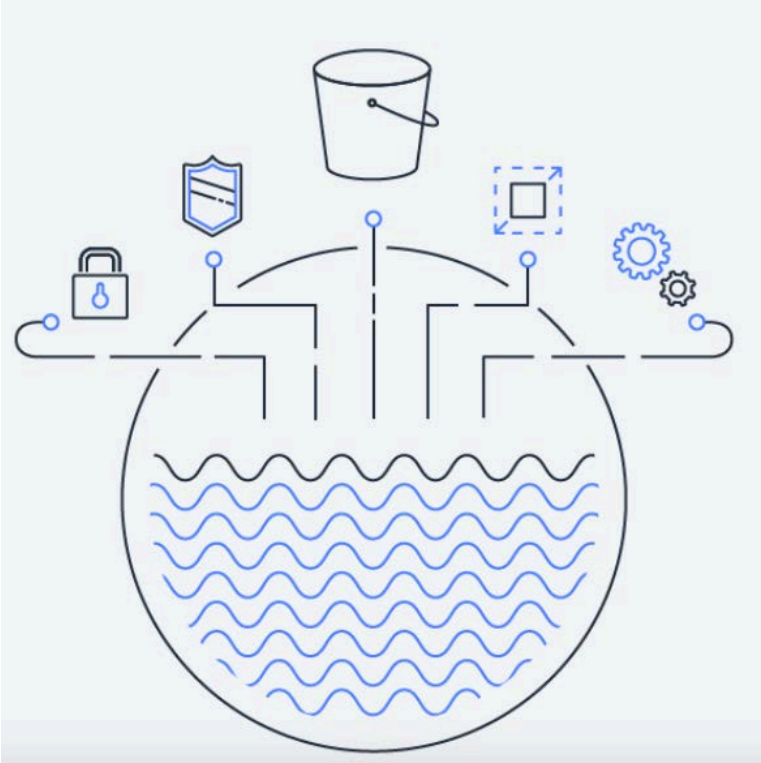

Fig 1. Benefits of building data lakes on AWS S3

Data Lakes on S3 can be used for various use cases such as Backup and to protect in-premises data lakes, modernizing data warehouses, running Big Data Analytics or using Advanced Machine learning.

HOW DATA LAKES HELP WITH BIG DATA CHALLENGES

The main challenges with Big Data in large organizations are:

1. Data Silos
2. Difficulty analyzing diverse datasets
3. Data controllership
4. Data security
5. Incorporating Machine Learning

Breaking down data silos

A major reason companies choose to create data lakes is to break down data silos. Having pockets of data in different places, controlled by different groups, inherently obscures data. This often happens when companies grow fast or acquire new business.

When this happens, its harder to make sense of data at an organizational or companywide level. It request manual data collection from many different sources. With so many teams operating independently, we loose efficiencies that could be achieved by solving problems together.

Its also harder to get granular details from the data, because not everybody has access to the various data repositories. For smaller queries, you could share a cut of the data in a spreadsheet, but challenges arise when data exceeds the capacity of a spreadsheet which happens very often at large companies. While, you can still get a high-level summary you wont be able to get granular details.

Data lakes solves this problem by uniting all the data to one central location. Teams can continue to function as nimble units but all reads lead back to the data lake for analytics. No more silos

Analyze diverse datasets

Another challenge of using different systems and approaches to data management is that the data structures and information vary.

If you wanted to combine all of this data in a traditional data warehouse without a data lake, it would require a lot of data preparation and export, transform and load or ETL operations. You would have to make tradeoffs on what to keep and what to lose and continually change the structure of a rigid system.

Data lakes allow you to import any amount of data in any format because there is no predefined schema. You can even ingest data in real time. You can collect data from multiple sources and move it into the data lake in its original format. You can also build links between information that might be labeled differently but represents the same thing.

Moving all your data to a data lake also improves what you can do with a traditional data warehouse. You have the flexibility to store highly structured, frequently accessed data in a data warehouse, while also keeping up to exabytes of structured, semi-structured and unstructured data in your data lake storage.

Managing Data Access

With data stored in so many locations, it’s difficult both to access all of it and to link to external tools for analysis. Large enterprises have data spread across multiple databases globally, with regional teams creating their own local version of datasets. That means

multiple access management credentials for some people. Many of the databases require access management support to do things such as change profiles or reset passwords. In addition, audits and controls must be in place for each database to ensure that nobody has improper access.

With a data lake, it's easier to get the right data to the right people at the right time. Instead of managing access for all the different locations in which data is stored, you only have to worry about one set of credentials. Data lakes have controls that allow authorized users to see, access, process or modify specific assets. Data lakes help ensure that unauthorized users are blocked from taking actions that would compromise data confidentiality and security.

Data is also stored in an open format, which makes it easier to work with different analytic services. The open format also makes it more likely for the data to be compatible with tools that don't even exist yet. Various roles in your organization, such as data scientists, data engineers, application developers and business analysts, can access data with their choice of analytic tools and frameworks.

In short, you're not locked in to a small set of tools, and a broader group of people can make sense of the data.

Accelerating machine learning

A data lake is a powerful foundation for machine learning and artificial intelligence), because they thrive on large, diverse datasets. Machine learning uses statistical algorithms that learn from existing data, a process called training, to make decisions about new data, a process called inference.

During training, patterns and relationships in the data are identified to build a model. The model allows you to make intelligent decisions about data it hasn't encountered before. The more data you have the better you can train your machine learning models, resulting in improved accuracy.

By moving all the data to a data lake, organizations can combine datasets to train and deploy more accurate models. Training machine learning models with more relevant data increases the accuracy of forecasting. In addition, it frees employees who were performing this task manually to work on more strategic projects, such as analyzing the forecasts to drive operations improvements in the field.

BUILDING DATA LAKES ON AWS

AWS Lake Formation is a service that makes it easy to set up a secure data lake in days. A data lake is a centralized, curated, and secured repository that stores all your data, both in its original form and prepared for analysis. A data lake enables you to break down data silos and combine different types of analytics to gain insights and guide better business decisions.

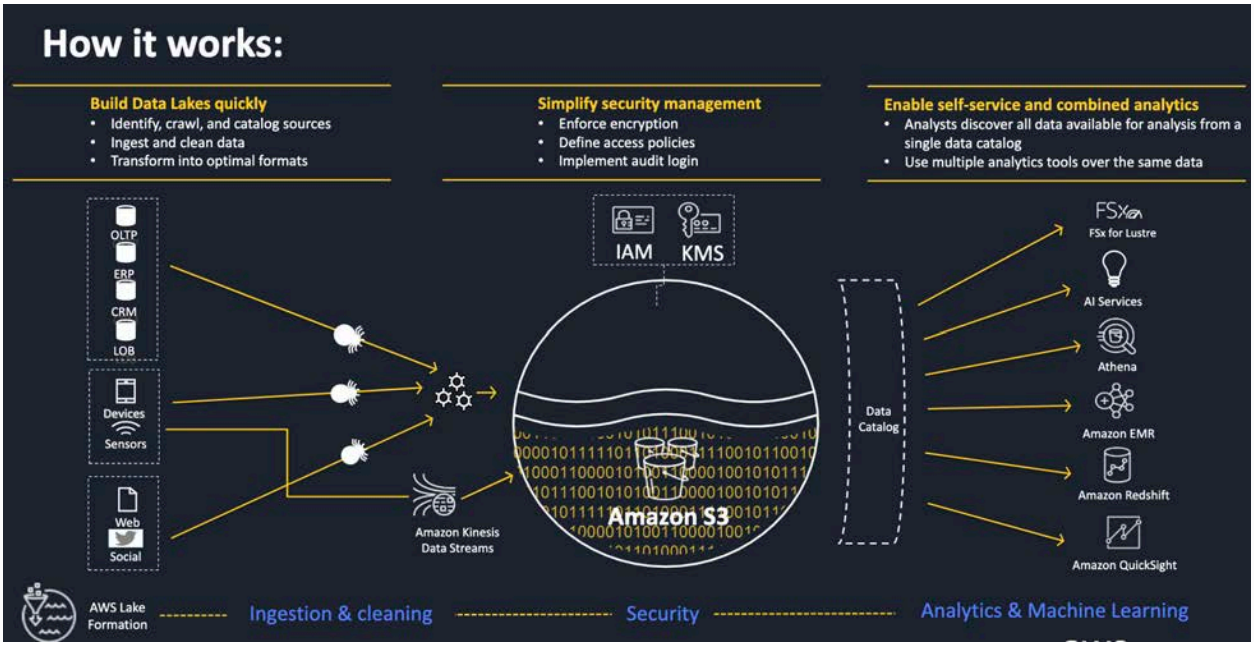

Fig 2: How a data lake works on AWS

However, setting up and managing data lakes today involves a lot of manual, complicated, and time-consuming tasks. This work includes loading data from diverse sources, monitoring those data flows, setting up partitions, turning on encryption and managing keys, defining transformation jobs and monitoring their operation, re-organizing data into a columnar format, configuring access control settings, deduplicating redundant data, matching linked records, granting access to data sets, and auditing access over time.

Creating a data lake with Lake Formation is as simple as defining where your data resides and what data access and security policies you want to apply. Lake Formation then collects and catalogs data from databases and object storage, moves the data into your new Amazon S3 data lake, cleans and classifies data using machine learning algorithms, and secures access to your sensitive data. Your users can then access a centralized catalog of data which describes available data sets and their appropriate usage. Your users then leverage these data sets with their choice of analytics and machine learning services, like Amazon EMR for Apache Spark, Amazon Redshift, Amazon Athena, Amazon Sagemaker, and Amazon QuickSight.

ACCESS / ANALYZE DATA FROM DATA LAKES

AWS Redshift

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It allows you to run complex analytic queries against petabytes of structured data using sophisticated query optimization, columnar storage on high-performance storage, and massively parallel query execution. Most results come back in seconds. With Redshift, you can start small for just $0.25 per hour with no commitments and scale out to petabytes of data for $1,000 per terabyte per year, less than a tenth the cost of traditional solutions. Amazon Redshift also includes Amazon Redshift Spectrum, allowing you to

directly run SQL queries against exabytes of unstructured data in Amazon S3 data lakes. No loading or transformation is required, and you can use open data formats, including Avro, CSV, Ion, JSON, ORC, Parquet, and more. Redshift Spectrum automatically scales query compute capacity based on the data being retrieved, so queries against Amazon S3 run fast, regardless of data set size.

Redshift Compatibility with BI Software

Amazon Redshift uses industry-standard SQL and is accessed using standard JDBC and ODBC drivers. You can download Amazon Redshift custom JDBC and ODBC drivers from the Connect Client tab of the Redshift Console. We have validated integrations with popular BI and ETL vendors, a number of which are offering free trials to help you get started loading and analyzing your data. You can also go to the AWS Marketplace to deploy and configure solutions designed to work with Amazon Redshift in minutes.

Redshift Spectrum supports all Amazon Redshift client tools. The client tools can continue to connect to the Amazon Redshift cluster endpoint using ODBC or JDBC connections. No changes are required.

You use exactly the same query syntax and have the same query capabilities to access tables in Redshift Spectrum as you have for tables in the local storage of your Redshift cluster. External tables are referenced using the schema name defined in the CREATE EXTERNAL SCHEMA command where they were registered.

DESIGN PATTERNS DATA LAKE ARCHITECTURE USING REDSHIFT

There are two common design patterns when moving data from source systems to a data warehouse. The primary difference between the two patterns is the point in the data-processing pipeline at which transformations happen. This also determines the set of tools used to ingest and transform the data, along with the underlying data structures, queries, and optimization engines used to analyze the data. The first pattern is ETL, which transforms the data before it is loaded into the data warehouse. The second pattern is ELT, which loads the data into the data warehouse and uses the familiar SQL semantics and power of the Massively Parallel Processing (MPP) architecture to perform the transformations within the data warehouse.

In the following diagram, the first represents ETL, in which data transformation is performed outside of the data warehouse with ETL tools. This pattern allows you to select your preferred tools for data transformations. The second diagram is ELT, in which the data transformation engine is built into the data warehouse for relational and SQL workloads. This pattern is powerful because it uses the highly optimized and scalable data storage and compute power of MPP architecture.

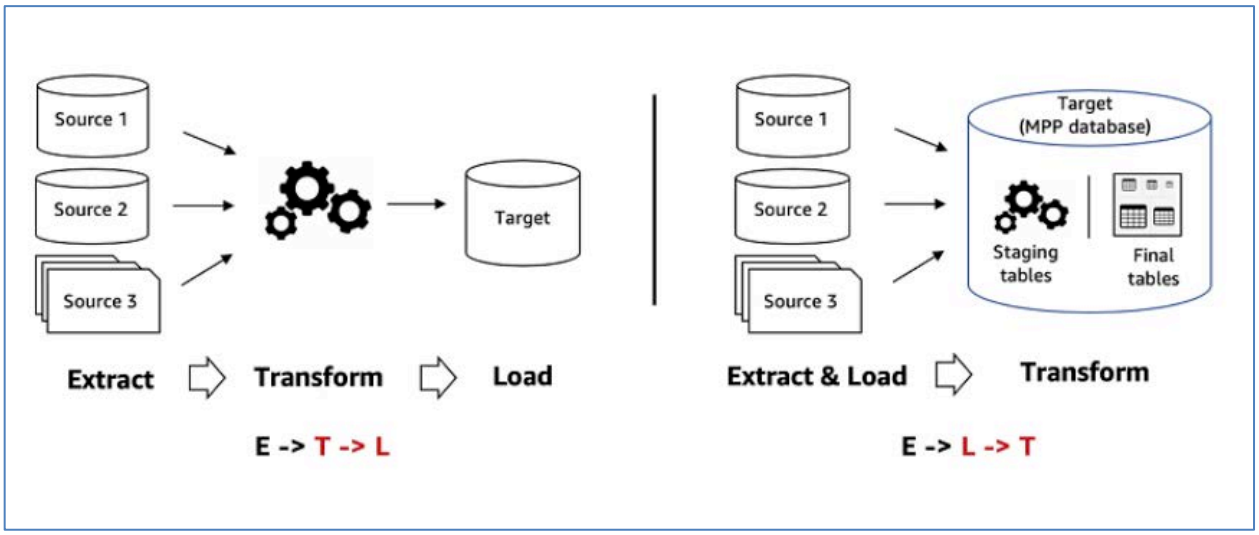

Fig 3: ETL vs ELT

REDSHIFT SPECTRUM

Amazon Redshift is a fully managed data warehouse service on AWS. It uses a distributed, MPP, and shared nothing architecture. Redshift Spectrum is a native feature of Amazon Redshift that enables you to run the familiar SQL of Amazon Redshift with the BI application and SQL client tools you currently use against all your data stored in open file formats in your data lake (Amazon S3).

A common pattern you may follow is to run queries that span both the frequently accessed *hot* data stored locally in Amazon Redshift and the *warm* or *cold* data stored cost-effectively in Amazon S3, using views with no schema binding for external tables. This enables you to independently scale your compute resources and storage across your cluster and S3 for various use cases.

Redshift Spectrum supports a variety of structured and unstructured file formats such as Apache Parquet, Avro, CSV, ORC, JSON to name a few. Because the data stored in S3 is in open file formats, the same data can serve as your single source of truth and other services such as Amazon Athena, Amazon EMR, and Amazon SageMaker can access it directly from your S3 data lake.

DATA LAKE EXPORT

Amazon Redshift now supports unloading the result of a query to your data lake on S3 in Apache Parquet, an efficient open columnar storage format for analytics. The Parquet format is up to two times faster to unload and consumes up to six times less storage in S3, compared to text formats. You can also specify one or more partition columns, so that unloaded data is automatically partitioned into folders in your S3 bucket to improve query performance and lower the cost for downstream consumption of the unloaded data. For example, you can choose to unload your marketing data and partition it by year, month, and day columns. This enables your queries to take advantage of partition pruning and skip scanning of non-relevant partitions when filtered by the partitioned columns, thereby improving query performance and lowering cost. For more information, see UNLOAD.

You have a requirement to unload a subset of the data from Amazon Redshift back to your data lake (S3) in an open and analytics-optimized columnar file format (Parquet). You then want to query the unloaded datasets from the data lake using Redshift Spectrum and other AWS services.

You have a requirement to share a single version of a set of curated metrics (computed in Amazon Redshift) across multiple business processes from the data lake. You can use ELT in Amazon Redshift to compute these metrics and then use the unload operation with optimized file format and partitioning to unload the computed metrics in the data lake.

You also have a requirement to pre-aggregate a set of commonly requested metrics from your end-users on a large dataset stored in the data lake (S3) cold storage using familiar SQL and unload the aggregated metrics in your data lake for downstream consumption. In other words, consider a batch workload that requires standard SQL joins and aggregations on a fairly large volume of relational and structured cold data stored in S3 for a short duration of time. You can use the power of Redshift Spectrum by spinning up one or many short-lived Amazon Redshift clusters that can perform the required SQL transformations on the data stored in S3, unload the transformed results back to S3 in an optimized file format, and terminate the unneeded Amazon Redshift clusters at the end of the processing. This way, you only pay for the duration in which your Amazon Redshift clusters serve your workloads.

As shown in the following diagram, once the transformed results are unloaded in S3, you then query the unloaded data from your data lake either using Redshift Spectrum if you have an existing Amazon Redshift cluster, Athena with its pay-per-use and serverless ad hoc and on-demand query model, or other BI & ELT tools such as SAS.

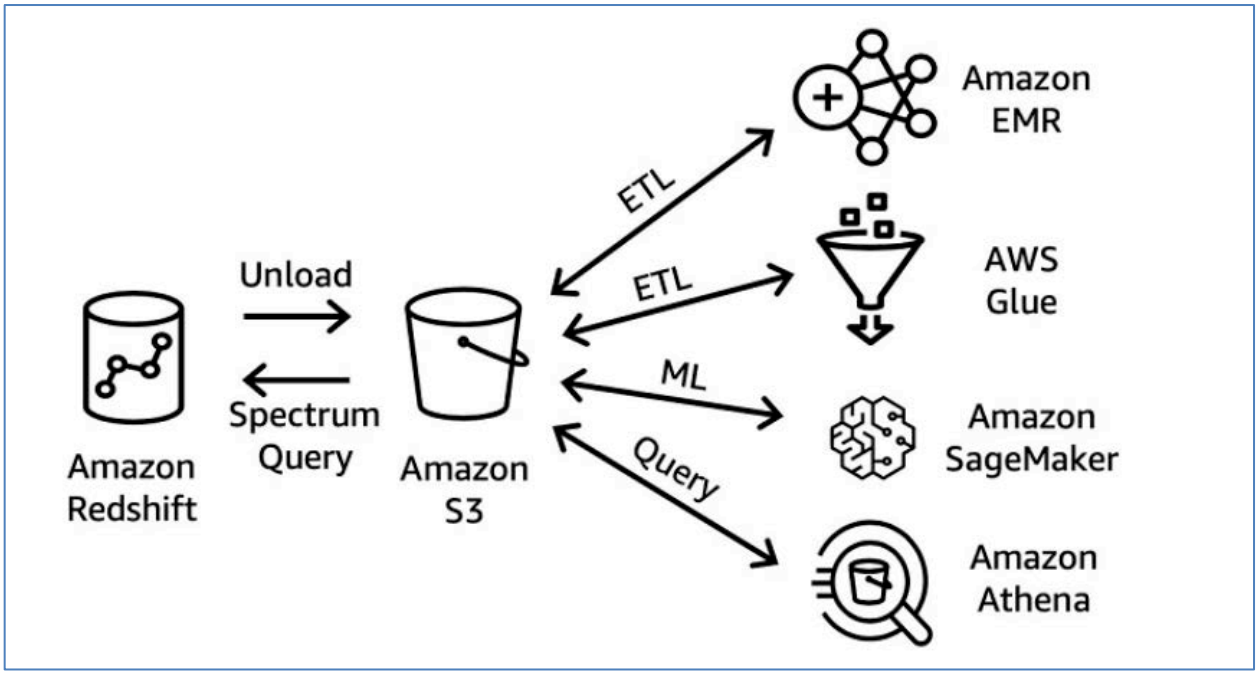

Fig 4 – ETL Architecture with Redshift and Redshift Spectrum

SAS ACCESS BULK LOAD/UNLOAD TO REDSHIFT

Beginning in SAS Viya 3.3, SAS/ACCESS Interface to Amazon Redshift includes SAS Data Connector to Amazon Redshift. The data connector enables you to load large amounts of data into the CAS server for parallel processing. The bulk load feature is an important step in loading data quickly to redshift using native Redshift support.

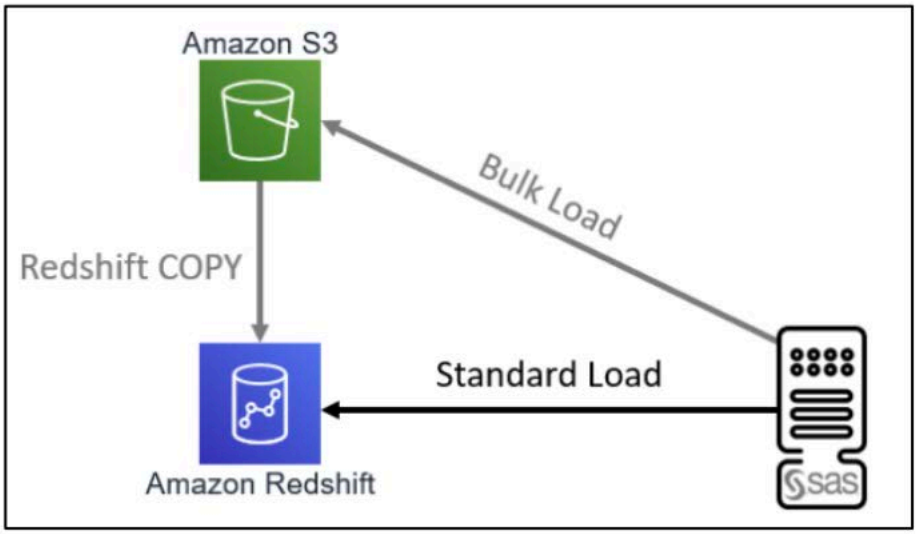

Fig 5 SAS Redshift Bulk Load

Code Snippet – Bulk Load

```
options sastrace=',,,ds' sastraceloc=saslog nostsuffix;

libname libred redshift server=rsserver db=rsdb user=myuserID pwd=myPwd
port=5439;

data libred.myclass(
   bulkload=yes
      bl_bucket=myBucket
      bl_key=99999
      bl_secret=12345
      bl_default_dir='/tmp'
      bl_region='us-east-1');
   set sashelp.class;
run;
```

Git Hub Code

AMAZON ATHENA

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to setup or manage, and you can start analyzing data immediately. You don't even need to load your data into Athena, it works directly with data stored in S3. To get started, just log into the Athena Management Console, define your schema, and start querying. Amazon Athena uses Presto with full standard SQL support and works with a variety of standard data formats, including CSV, JSON, ORC, Apache Parquet and Avro. While Amazon Athena is ideal for quick, ad-hoc querying and integrates with SAS Studio for easy visualization, it can also handle complex analysis, including large joins, window functions, and arrays.

Amazon Athena uses Presto with full standard SQL support and works with a variety of standard data formats, including CSV, JSON, ORC, Avro, and Parquet. Athena can handle complex analysis, including large joins, window functions, and arrays. Because Amazon Athena uses Amazon S3 as the underlying data store, it is highly available and durable with data redundantly stored across multiple facilities and multiple devices in each facility

Querying using Athena

Amazon Athena comes with an ODBC and JDBC driver that you can use with other business intelligence tools and SQL clients. You can access Amazon Athena using the AWS Management Console, the Amazon Athena API, or the AWS CLI.
You can improve the performance of your query by compressing, partitioning, or converting your data into columnar formats. Amazon Athena supports open source columnar data formats such as Apache Parquet and Apache ORC. Converting your data into a compressed, columnar format lowers your cost and improves query performance by enabling Athena to scan less data from S3 when executing your query.

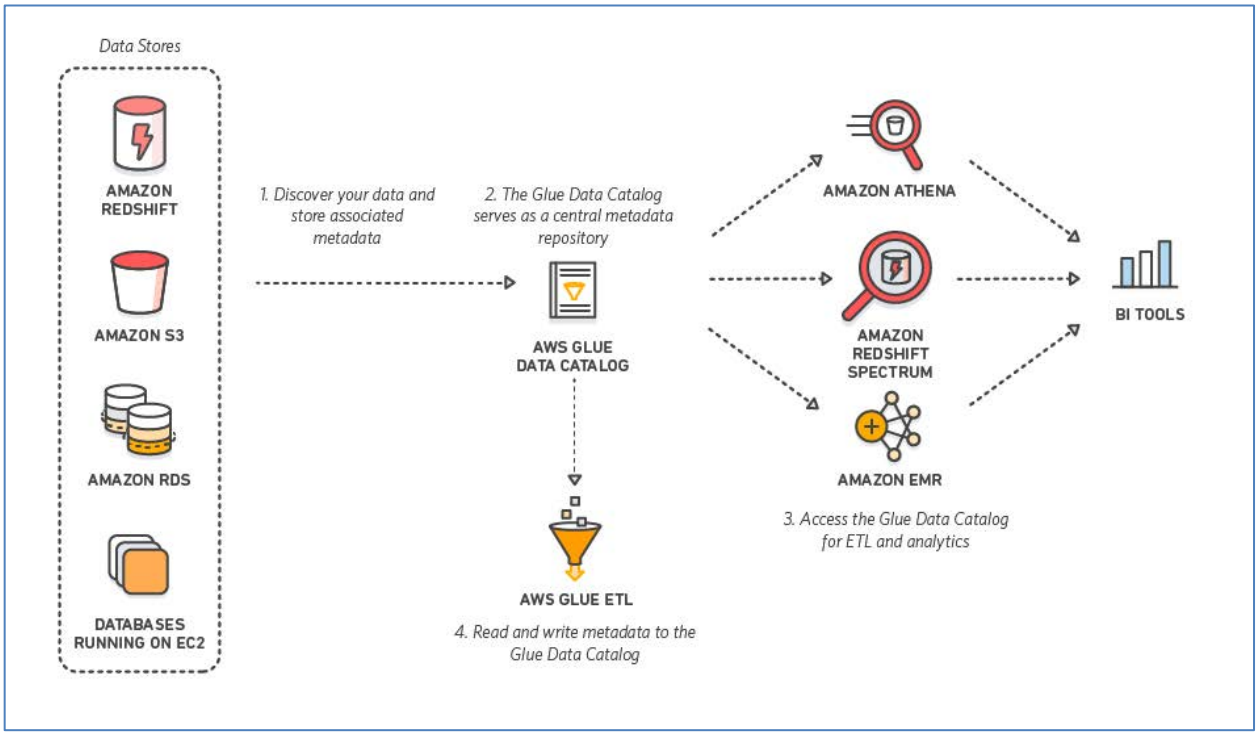

Fig 6: Querying data from a data lake using Athena

S3 SELECT

With Amazon S3 Select, you can use simple structured query language (SQL) statements to filter the contents of Amazon S3 objects and retrieve just the subset of data that you need. By using Amazon S3 Select to filter this data, you can reduce the amount of data that Amazon S3 transfers, which reduces the cost and latency to retrieve this data.

Amazon S3 Select works on objects stored in CSV, JSON, or Apache Parquet format. It also works with objects that are compressed with GZIP or BZIP2 (for CSV and JSON objects only), and server-side encrypted objects. You can specify the format of the results as either CSV or JSON, and you can determine how the records in the result are delimited.

You pass SQL expressions to Amazon S3 in the request. Amazon S3 Select supports a subset of SQL. For more information about the SQL elements that are supported by Amazon S3 Select, see SQL Reference for Amazon S3 Select and S3 Glacier Select.

You can perform SQL queries using AWS SDKs, the SELECT Object Content REST API, the AWS Command Line Interface (AWS CLI), or the Amazon S3 console. The Amazon S3 console limits the amount of data returned to 40 MB. To retrieve more data, use the AWS CLI or the API.

CONCLUSION

In this paper, readers get an overview of the various services that are used build a data lake at AWS. With its many benefits, S3 is main building block for Data Lakes in AWS and readers get a view of these benefits such as reliability, security and durability with S3. A data lake can be setup using Lake Formation and readers would get an indepth view into how data lake can be setup quickly. Readers also got to understand the key capabilities of Redshift and Redshift spectrum used as a datawarehouse and how SAS users can effectively access the same. Readers were further exposed to other analytics tools such as Athena and S3 select using pay per query pricing which would further add access opportunities to a data lake. This paper has provided users a comprehensive view into multiple AWS services and how SAS users benefit from each of these services to access data lakes.

RECOMMENDED READING

- Redshift ETL/ELT Design Patterns
- Data Lake Storage
- SAS Athena
- Lake Formation FAQs
- Best practices whitepaper

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dilip Rajan
Amazon Web Services
rajand@amazon.com