International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021 ISSN No: 2348-9510

DATA MIGRATION FROM ON-PREMISES STORAGE TO AWS AND SNOWFLAKE USING AWS S3 AND AWS GLUE: TOOLS, TECHNIQUES, AND BEST PRACTICES

Ravi Kiran Koppichetti
koppichettiravikiran@gmail.com

Abstract

Cloud data migration is a process of moving data, services, and applications from On-premises to distributed cloud computing infrastructure. One of the crucial steps of this process is moving locally stored data to a public cloud provider. There are three main ways to migrate data flow from local storage into a cloud solution, such as Amazon S3, and multiple steps and checks before loading data into Snowflake for analytical capabilities. This paper provides a road map to migrate data from on-premises systems, such as the Network File System (NFS) and object storage, to Snowflake using Amazon S3 and AWS Glue. It also addresses key considerations and challenges and provides insights into best practices to streamline the migration process.

Keywords : Data migration, Cloud computing, Amazon S3, Snowflake, AWS Glue, On-premises

I. INTRODUCTION

For many organizations that either want to move away from slow legacy systems or consolidate data from various source systems for an analytical process, data migration is a critical step for them. The data migration companies perform could result in cloud adoption and system modernization, enhanced organization security and operational efficiency, or both. Many organizations moving to the cloud utilize its scalability, flexibility, and cost-effectiveness features. Most companies prefer AWS and Snowflake cloud platforms for their comprehensive data storage, security, processing, and analytics environment.

The data migration process from On-premises to Cloud platforms is complex. It involves multiple stages of transferring, storing, retrieving, and transforming large amounts of data based on a defined business logic. During this process, it is important to ensure data integrity and security. This paper aims to examine how tools offered by AWS, such as Amazon S3 and AWS Glue, can facilitate data migration from On-premises to Snowflake for analytical reporting.

II. OVERVIEW OF KEY AWS SERVICES

A. Amazon S3 (Simple Storage Service)

Amazon Simple Storage Service (Amazon S3) is a cloud-native object-based storage solution that allows organizations to load, store, and access huge amounts of data with reliability and scalability. At its crux, S3 enables users to store objects of any type and size while providing easy access through a web interface (S3 Console) and the AWS Command Line Interface (CLI) [1].

S3's object-based architecture allows users to store everything from small documents to large data

lakes. With data centers spread across the globe and built-in backup systems, S3 ensures that we can always access our information, regardless of our needs. S3 is a flexible service from Amazon that can store anything from infrequent backups from organizations to archival data storage. Amazon S3 offers encryption features, access control policies, and logging capabilities that give organizations tremendous confidence in securely storing sensitive data [3].

The three main features of Amazon S3 are:

1. **Scalability and Durability:** Amazon S3 is designed for virtually unlimited scalability and provides 99.999999999% (11 9's) durability. This durability makes it highly reliable for storing large amounts of data, ensuring availability and resiliency [1].

2. **Data Management and Security**: Amazon S3 provides a range of data management and security features, including access control through AWS Identity and Access Management (IAM), encryption (both in-transit and at-rest), and object tagging. These features allow users to organize data and enforce fine-grained access policies [1].

3. **Cost-Effective Storage Classes**: S3 offers multiple storage classes, including S3 Standard, S3 Intelligent-Tiering, S3 Glacier, and S3 Glacier Deep Archive, which cater to different use cases and retrieval speeds. These classes allow customers to optimize costs by selecting storage options based on data access frequency and required retrieval time [1]

B. AWS Glue

AWS Glue is a fully managed ETL (Extract, Transform, Load) service provided by Amazon Web Services designed to prepare and transform data for analytics and machine learning. It automates much of the work involved in data integration by handling tasks like data discovery, schema inferencing, job scheduling, and metadata management. These features help a data engineer to set up and manage data pipelines easily [1,4].

Essential features of AWS Glue include:

1. **Data Catalog:** AWS Glue's central catalog automatically discovers and stores metadata about our data sources, such as schemas and classifications, to make data more accessible to organize and search [1].

2. **Serverless ETL**: Glue is serverless, so we do not have to manage infrastructure. We create and run ETL jobs, and Glue scales the resources to handle the job's needs [1].

3. **Integrations:** AWS Glue integrates with various AWS services, such as Amazon S3, RDS, Redshift, and Athena, making it easier to manage data across the AWS cloud. It can also integrate directly with Snowflake allowing developers to create seamless pipelines for data loading and transformations. AWS Glue comes with a built-in connector for Snowflake which helps us move data from S3 into Snowflake [1,2].

III. DATA MIGRATION WORKFLOW FROM ON-PREMISES STORAGE TO AWS AND SNOWFLAKE

A. Step 1: Preparing Data on On-Premises Storage

Before migrating data files to AWS, the data on the on-premises systems should be appropriately

organized and prepared for transfer:

1. **Data Assessment:** The process of data assessment involves evaluating the size and schema of the data and if any transformations required at the source. All structured, semi-structured, and unstructured data types can be handled in this data migration process [5].

2. **Data Cleansing:** Before initiating the migration process, we need to, if needed, remove duplicates and handle null values and inconsistencies to make sure the data is accurate, consistent and has no errors [5].

3. **Compression and Partitioning**: When migrating large data lakes, the data can be compressed and partitioned into smaller fragments. This process improves the performance of data transfer and transformation process in later steps [5].

B. Step 2: Data Transfer to Amazon S3

After preparing the data for migration, it is transferred to an Amazon S3 bucket for staging before processing and moving it into Snowflake. Below are the ways to migrate data from On-Premises storage to AWS S3:

1. **Using AWS Snowball:** If the dataset is large and the network bandwidth between On-premises and AWS is limited, AWS Snowball can physically transport data from on-premises systems to the AWS data center. Snowball appliances are shipped to the organization to be filled with data and returned to AWS for upload to S3 [1].

2. **Using AWS Direct Connect**: For organizations with fast and reliable networks, AWS Direct Connect can provide a devoted network link between on-premises storage and S3, enabling faster and more reliable data migration [1,6].

3. **Using AWS DataSync**: For organizations that need to automate large-scale data transfers and at recurrence, AWS DataSync is a reliable method. It also allows encryption, validation, scheduling, and tracking, making it best for continuous data migration, disaster recovery, and synchronization [1].

C. Step 3: Data Transformation Using AWS Glue

After we transfer the data files to Amazon S3, we use AWS Glue to perform data transformations. Below are the steps to transform data using AWS Glue:

1. **Data Cataloging:** AWS Glue automatically discovers the data in S3 using its Data Catalog feature and organizes it into tables and schemas. This metadata repository helps manage data and its transformations.

2. **ETL/ ELT Jobs**: Users can create ETL (Extract, Transform, Load) or ELT (Extract, Load, Transform) jobs to perform data transformations using AWS Glue. These transformations may include:

- Data cleansing: Handling missing or incorrect data.
- Schema conversion: Mapping the on-premises table schema to the target Snowflake table schema.

- Data enrichment: Processing the source data for analytics.

3. **Automating the ETL/ELT job**: Users can run the ETL/ELT jobs they defined and developed either on-demand or on a schedule, enabling batch processing and real-time data migration.

D. Step 4: Loading Data into Snowflake

After transforming the On-premises data, which is available in AWS S3, using AWS Glue, data is ready for loading into Snowflake for analysis or storing. Below are the steps to migrate data into Snowflake:

1. **Direct Loading via AWS Glue**: AWS Glue provides built-in connectors for Snowflake, allowing users to load data directly from S3 into Snowflake. The data can be loaded into Snowflake tables using the COPY INTO command or through Snowpipe for continuous loading [1,2].

2. **Optimizing Data Loading:** During the loading process, the data is stored in columnar format (Parquet, ORC) in S3, which is highly compatible with Snowflake. Snowflake supports automatic clustering for large datasets to improve query performance post-migration [1,2].

3. **Data Verification**: After the data is loaded into Snowflake, data validation steps should be performed to ensure that it was migrated correctly and is accessible for queries.

IV. KEY CONSIDERATIONS AND CHALLENGES

A. **Data Security:** During a data migration, data can either be in transit or at rest. It is crucial to encrypt sensitive data throughout the process. AWS Key Management Services (KMS) and Snowflake's end-to-end encryption have the capabilities to ensure data protection during data migration. Additional AWS and Snowflake provide IAM and Access roles to secure access controls on data [1,2,7].

B. **Performance Optimization**: Using the techniques presented in this guide, we can transfer data of any size from On-premises storage to Snowflake. To increase the performance of the data migration, it is vital to partition data into smaller fragments within Amazon S3 and enable AWS Glue to process the data in parallel on AWS S3 files. During data migration and transformations, we can also use query optimization and materialized views of Snowflake to optimize performance [1,2,8].

C. Cost Management: AWS data transfer, storage, and computing costs for running AWS Glue jobs and Snowflake queries can add up very quickly. Monitoring and optimizing usage using AWS Cost Explorer and Snowflake's Resource Monitors is essential [1,2,9].

V. BEST PRACTICES FOR DATA MIGRATION

A. **Incremental Data Migration**: The user should start with a small batch of data for pilot migration to test the ETL/ELT pipelines and transformation logic implemented and identify potential issues early in the process.

B. **Automated Data Processing**: AWS Glue can automatically discover and transform data with minimal manual intervention. The user can highly utilize this feature to help with data migration and transformation.

C. **Testing and Validation**: After migrating a small batch of data during the pilot batch, the user should rigidly test the migrated data to ensure accuracy and consistency per business requirements. Even during production data migration activities, the users should use the AWS Glue job monitoring feature to track the success and failure of data processing tasks and take action accordingly.

D. Documentation: The user should always maintain highly detailed documentation of the migration process, which should contain details of data sources, transformations, schemas, test scripts, test cases, and business use-case to ensure a smooth handoff and ongoing data maintenance.

VI. CONCLUSION

The data migration process from on-premises data storage to AWS S3 and Snowflake to realize analytical capabilities can be complex but rewarding. By taking advantage of AWS S3 for scalable storage, AWS KMS for encryption, and AWS Glue for ETL/ETL/ Streaming capabilities, organizations can successfully migrate large volumes of data or small batches at higher intervals of data to the cloud while minimizing disruptions and ensuring data integrity. The combination of AWS and Snowflake provides robust data storage, security, processing, and analytical capabilities, and allowing businesses to recognize the full potential of their data in the cloud.

REFERENCES

1. Amazon Web Services, Inc. "Amazon Web Services Documentation." [Online]. Available: https://docs.aws.amazon.com. [Accessed: Jan. 18, 2021].
2. Snowflake Inc., "Snowflake: The Data Cloud," [Online]. Available: https://www.snowflake.com. [Accessed: Jan. 14, 2021].
3. M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska, "Building a database on S3," in Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08), New York, NY, USA, 2008, pp. 251–264. doi: https://doi.org/10.1145/1376616.1376645.
4. S. Gentilini, Leveraging AWS Glue to Implement Spark-based ETL Jobs for a Data Warehousing Solution, 2020. [Online]. Available: https://hdl.handle.net/20.500.14239/13767. [Accessed: Feb. 12, 2021].
5. N. G. Lakshmi, "Database Migration on Premises to AWS RDS," EAI Endorsed Transactions on Cloud Systems, vol. 3, (11), 2018. Available: https://login.cyrano.ucmo.edu/login?url=https://www.proquest.com/scholarly-journals/database-migration-on-premises-aws-rds/docview/2305560884/se-2. DOI: https://doi.org/10.4108/eai.11-4-2018.154463.
6. T. Wubu, Migration of Traditional IT System to Cloud Computing with Amazon Web Services, 2020.
7. S. Narula, A. Jain and Prachi, "Cloud Computing Security: Amazon Web Service," 2015 Fifth International Conference on Advanced Computing & Communication Technologies, Haryana, India, 2015, pp. 501-505, doi: 10.1109/ACCT.2015.20.
8. A. Das, S. Imai, S. Patterson and M. P. Wittie, "Performance Optimization for Edge-Cloud

Serverless Platforms via Dynamic Task Placement," 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, VIC, Australia, 2020, pp. 41-50, doi: 10.1109/CCGrid49817.2020.00-89.

9. J. Baron and S. Kotecha, Storage Options in the AWS Cloud, Amazon Web Services, Washington DC, Tech. Rep., 2013.