IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, July 2023

Orchestrate Redshift ETL using AWS Glue and Step Functions

Aditya Chouhan, Khushi Soni, Nikhil Soni, Prof. Manoj Kumar Gupta
Department of CSIT
Acropolis Institute of Technology & Research, Indore, M.P., India

Abstract: *Data Analytics and Machine Learning work-streams rely on ETL for their basis. ETL cleanses and organizes data using a set of business rules to meet particular business intelligence requirements, such as monthly reporting. Still, it may also handle complex analytics to enhance back-end operations or end-user experiences. An organization's ETL is frequently used to Retrieve data from legacy systems, To improve data accuracy and reliability, clean the data. Update a target database with data*

Keywords: ETL tools, Data integration, Amazon redshift

I. INTRODUCTION

AWS is a cloud computing platform that provides a broad range of services that allow users to deploy, operate, and manage applications. AWS offers a variety of services, including compute, storage, networking, and application services. AWS also offers a variety of tools and services that allow users to manage their applications and data.

Redshift Orchestrate is a tool that makes it easy to manage and orchestrate your redshift clusters. It provides a centralized interface for managing your clusters, monitoring their health, and managing your dataflows. It provides a single interface to manage your cluster, including provisioning, monitoring, and scaling. It also provides a rich set of features to help you manage your data. For example, you can use Redshift Orchestrate to manage your cluster, optimize your data, and manage your costs. It also provides a rich set of features to help you manage your data.

Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the AWS Cloud. An Amazon Redshift data warehouse is a collection of computing resources called nodes, which are organized into a group called a cluster. Each cluster runs an Amazon Redshift engine and contains one or more databases.

The Amazon Redshift service manages all of the work of setting up, operating, and scaling a data warehouse. These tasks include provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine.

Cluster management: An Amazon Redshift cluster is a set of nodes, which consists of a leader node and one or more compute nodes. The type and number of compute nodes that you need depends on the size of your data, the number of queries you will run, and the query execution performance that you need.

Creating and managing clusters: Depending on your data warehousing needs, you can start with a small, single-node cluster and easily scale up to a larger, multi-node cluster as your requirements change. You can add or remove compute nodes to the cluster without any interruption to the service. For more information, see Amazon Redshift clusters.

Reserving compute nodes

If you intend to keep your cluster running for a year or longer, you can save money by reserving compute nodes for a one-year or three-year period. Reserving compute nodes offers significant savings compared to the hourly rates that you pay when you provision compute nodes on demand. For more information, see Purchasing Amazon Redshift reserved nodes.

Creating cluster snapshots: Snapshots are point-in-time backups of a cluster. There are two types of snapshots: automated and manual. Amazon Redshift stores these snapshots internally in Amazon Simple Storage Service (Amazon S3) by using an encrypted Secure Sockets Layer (SSL) connection. If you need to restore from a snapshot, Amazon Redshift creates a new cluster and imports data from the snapshot that you specify. For more information about snapshots, see Amazon Redshift snapshots and backups.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, July 2023

Cluster access and security: There are several features related to cluster access and security in Amazon Redshift. These features help you to control access to your cluster, define connectivity rules, and encrypt data and connections. These features are in addition to features related to database access and security in Amazon Redshift. For more information about database security, see Managing Database Security in the *Amazon Redshift Database Developer Guide*.

AWS accounts and IAM credentials: By default, an Amazon Redshift cluster is only accessible to the AWS account that creates the cluster. The cluster is locked down so that no one else has access. Within your AWS account, you use the AWS Identity and Access Management (IAM) service to create users that assume IAM roles with permissions attached to control cluster operations. For more information, see Security in Amazon Redshift.

Security groups: By default, any cluster that you create is closed to everyone. IAM credentials only control access to the Amazon Redshift API-related resources: the Amazon Redshift console, command line interface (CLI), API, and SDK. To enable access to the cluster from SQL client tools via JDBC or ODBC, you use security groups:

If you are using the EC2-VPC platform for your Amazon Redshift cluster, you must use VPC security groups. We recommend that you launch your cluster in an EC2-VPC platform.

You cannot move a cluster to a VPC after it has been launched with EC2-Classic. However, you can restore an EC2-Classic snapshot to an EC2-VPC cluster using the Amazon Redshift console. For more information, see Restoring a cluster from a snapshot.

If you are using the EC2-Classic platform for your Amazon Redshift cluster, you must use Amazon Redshift security groups.

In either case, you add rules to the security group to grant explicit inbound access to a specific range of CIDR/IP addresses or to an Amazon Elastic Compute Cloud (Amazon EC2) security group if your SQL client runs on an Amazon EC2 instance. For more information, see Amazon Redshift cluster security groups.

In addition to the inbound access rules, you create database users to provide credentials to authenticate to the database within the cluster itself. For more information, see Databases in this topic.

Encryption: When you provision the cluster, you can optionally choose to encrypt the cluster for additional security. When you enable encryption, Amazon Redshift stores all data in user-created tables in an encrypted format. You can use AWS Key Management Service (AWS KMS) to manage your Amazon Redshift encryption keys.

Encryption is an immutable property of the cluster. The only way to switch from an encrypted cluster to a cluster that is not encrypted is to unload the data and reload it into a new cluster. Encryption applies to the cluster and any backups. When you restore a cluster from an encrypted snapshot, the new cluster is encrypted as well.

For more information about encryption, keys, and hardware security modules, see Amazon Redshift database encryption.

SSL connections: You can use Secure Sockets Layer (SSL) encryption to encrypt the connection between your SQL client and your cluster. For more information, see Configuring security options for connections.

Monitoring clusters: There are several features related to monitoring in Amazon Redshift. You can use database audit logging to generate activity logs, configure events and notification subscriptions to track information of interest,. Use the metrics in Amazon Redshift and Amazon CloudWatch to learn about the health and performance of your clusters and databases.

Database audit logging: You can use the database audit logging feature to track information about authentication attempts, connections, disconnections, changes to database user definitions, and queries run in the database. This information is useful for security and troubleshooting purposes in Amazon Redshift. The logs are stored in Amazon S3 buckets. For more information, see Database audit logging.

Events and notifications: Amazon Redshift tracks events and retains information about them for a period of several weeks in your AWS account. For each event, Amazon Redshift reports information such as the date the event occurred, a description, the event source (for example, a cluster, a parameter group, or a snapshot), and the source ID. You can create Amazon Redshift event notification subscriptions that specify a set of event filters. When an event occurs that matches the filter criteria, Amazon Redshift uses Amazon Simple Notification Service to actively inform you that the event has occurred. For more information about events and notifications, see Amazon Redshift events.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, July 2023

Performance: Amazon Redshift provides performance metrics and data so that you can track the health and performance of your clusters and databases. Amazon Redshift uses Amazon CloudWatch metrics to monitor the physical aspects of the cluster, such as CPU utilization, latency, and throughput. Amazon Redshift also provides query and load performance data to help you monitor the database activity in your cluster. For more information about performance metrics and monitoring, see Monitoring Amazon Redshift cluster performance.

Databases: Amazon Redshift creates one database when you provision a cluster. This is the database you use to load data and run queries on your data. You can create additional databases as needed by running a SQL command. For more information about creating additional databases, go to Step 1: Create a database in the *Amazon Redshift Database Developer Guide*.

When you provision a cluster, you specify an admin user who has access to all of the databases that are created within the cluster. This admin user is a superuser who is the only user with access to the database initially, though this user can create additional superusers and users. For more information, go to Superusers and Users in the *Amazon Redshift Database Developer Guide*.

Amazon Redshift uses parameter groups to define the behavior of all databases in a cluster, such as date presentation style and floating-point precision. If you don't specify a parameter group when you provision your cluster, Amazon Redshift associates a default parameter group with the cluster. For more information, see Amazon Redshift parameter groups.

1.1 Problem Statement:

1. ETL cleanses and organizes data using a set of business rules to meet particular business intelligence requirements.
1. In this project, we use in-house AWS tools to orchestrate end-to-end loading and deriving business insights. Since it uses in-house tools, the availability and durability of the solution are guaranteed by AWS.

1.2 Objective

- Extract, Transform, and Load, or ETL, is a data integration process that integrates data from various sources into a single, consistent data store put into a data warehouse or other destination system.
- Data Analytics and Machine Learning work-streams rely on ETL for their basis.
- ETL cleanses and organizes data using a set of business rules to meet particular business intelligence requirements, such as monthly reporting.

II. PROCEDURE

The steps in this process are as follows:

The state machine launches a series of runs of an AWS Glue Python Shell job (more on how and why I use a single job later in this post!) with parameters for retrieving database connection information from AWS Secrets Manager and an .sql file from S3.

Each run of the AWS Glue Python Shell job uses the database connection information to connect to the Amazon Redshift cluster and submit the queries contained in the .sql file.

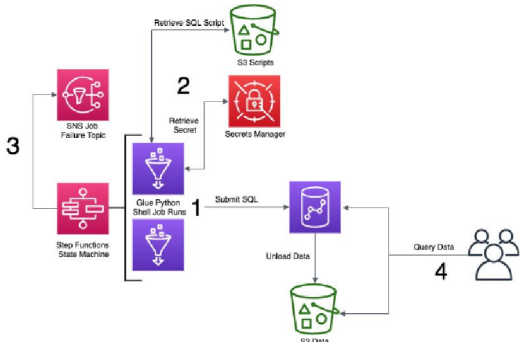
For Task 1: The cluster utilizes Amazon Redshift Spectrum to read data from S3 and load it into an Amazon Redshift table. Amazon Redshift Spectrum is commonly used as an means for loading data to Amazon Redshift. (See Step 7 of *Twelve Best Practices for Amazon Redshift Spectrum* for more information.)

For Task 2: The cluster executes an aggregation query and exports the results to another Amazon S3 location via UNLOAD.

The state machine may send a notification to an Amazon Simple Notification Service (SNS) topic in the case of pipeline failure.

Users can query the data from the cluster and/or retrieve report output files directly from S3.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, July 2023



III. METHODOLOGY

Modern data lakes depend on extract, transform, and load (ETL) operations to convert bulk information into usable data. This post walks through implementing an ETL orchestration process that is loosely coupled using AWS Step function, AWS Lambda and AWS Batch to target an Amazon Redshift cluster.

Because Amazon Redshift uses columnar storage, it is well suited for fast analytical insights using the convenient ANSI SQL queries. You can rapidly scale your Amazon Redshift clusters up and down in minutes to meet the demanding workloads for both your end-user reports and timely data refresh into the data warehouse.

AWS Step Functions makes it easy to develop and use repeatable workflows that scale well. Step Functions lets you build automation workflows from individual Lambda functions. Each function performs a discrete task and lets you develop, test, and modify the components of your workflow quickly and seamlessly.

An ETL process refreshes your data warehouse from source systems, organizing the raw data into a format you can more readily use. Most organizations run ETL as a batch or as part of a real-time ingest process to keep the data warehouse current and provide timely analytics. A fully automated and highly scalable ETL process helps minimize the operational effort that you must invest in managing the regular ETL pipelines. It also ensures the timely and accurate refresh of your data warehouse. You can tailor this process to refresh data into any data warehouse or the data lake.

This post also provides an AWS cloud formation template that launches the entire sample ETL process in one click to refresh the TPC-DS dataset. Find the template link in the *Set up the entire workflow using AWS Cloud Formation* section.

IV. TECHNOLOGY

Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 200 AWS services are available. New services can be provisioned quickly, without the upfront fixed expense. This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements. This whitepaper provides you with an overview of the benefits of the AWS Cloud and introduces you to the services that make up the platform.

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. It is optimized for datasets ranging from a few hundred gigabytes to a petabyte or more and costs less than $1,000 per terabyte per year, a tenth the cost of most traditional data warehousing solutions.

What they can do?

Regardless of skill level, get started in a few clicks. Power decisions by analyzing data across data warehouses, operational databases, and data lakes. Create and train ML models using familiar SQL. Securely share data across departments or regions. Build data systems and applications with as much flexibility and granular controls as required.

Analyze all your data

Easy, secure, reliable

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, July 2023

V. LITERATURE REVIEW

The state machine starts a series of AWS Glue Python Shell jobs, each with parameters for obtaining database connection information from AWS Secrets Manager and a SQL file from S3.The database connection information is used by each execution of the AWS Glue Python Shell task to connect to the Amazon Redshift cluster and submit the queries in the SQL file.The cluster utilizes Amazon Redshift Spectrum to read data from S3 and load it into an Amazon Redshift table. The cluster executes an aggregation query and exports the results to another Amazon S3 location via UNLOAD. The state machine may send a notification to an Amazon Simple Notification Service (SNS) topic in the case of pipeline failure. Users can query the data from the cluster or retrieve report output files directly from S3/Redshift using QuickSight.

VI. ACKNOWLEDGEMENTS

In the first place, we would like to thank all of the engineers who worked so hard to complete the initial edition of Amazon Redshift and who are still pushing innovation at a rapid rate while upholding the operational perfection of the service for our clients. The development of the database engine by the ParAccel team, which is currently at Actian, which in turn benefited from work by the PostgreSQL community, is the ancestor of Amazon Redshift. Their efforts greatly accelerated the release of Amazon Redshift. We also thank all of the reviewers for their insightful criticism and Prof. Manoj Kumar Gupta , Prof. Nisha Rathi for guiding this paper.

REFERENCES

[1]. Amazon Redshift and the Case for Simpler Data Warehouses Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, Vidhya Srinivasan.
[2] Daniel J. Abadi, Samuel R. Madden, and Miguel Ferreira. Integrating compression and execution in column-oriented database systems. In Proceedings of the ACM SIGMOD Conference on Management of Data, pages 671–682, 2006.
[3] https://aws.amazon.com/blogs/big-data/orchestrate-amazon-redshift-based-etl-workflows-with-aws-step-functions-and-aws-glue/
[4] https://docs.aws.amazon.com/redshift/latest/mgmt/overview.html