AWS RedShift

Amazon Redshift is a fully managed, petabyte-scale data warehouse service. With Amazon Redshift, you can query petabytes of structured and semi-structured data across your data warehouse and your data lake using standard SQL.

AWS Glue is a fully managed ETL service that makes it easier to prepare and load data for analytics. AWS Glue discovers your data and stores the associated metadata (for example, table definitions and schema) in the AWS Glue Data Catalog. Your cataloged data is immediately searchable, can be queried, and is available for ETL.

Amazon Redshift is not part of the AWS Free Tier, but it does offer a free trial for new users.
If you're new to Amazon Redshift, you can access a free trial that includes:

- **Amazon Redshift Serverless**: A $300 credit valid for 90 days, which can be used toward your compute and usage costs.
- **Provisioned Clusters**: A two-month trial offering 750 free hours per month of the `dw2.large` node type, which provides 160 GB of compressed SSD storage.

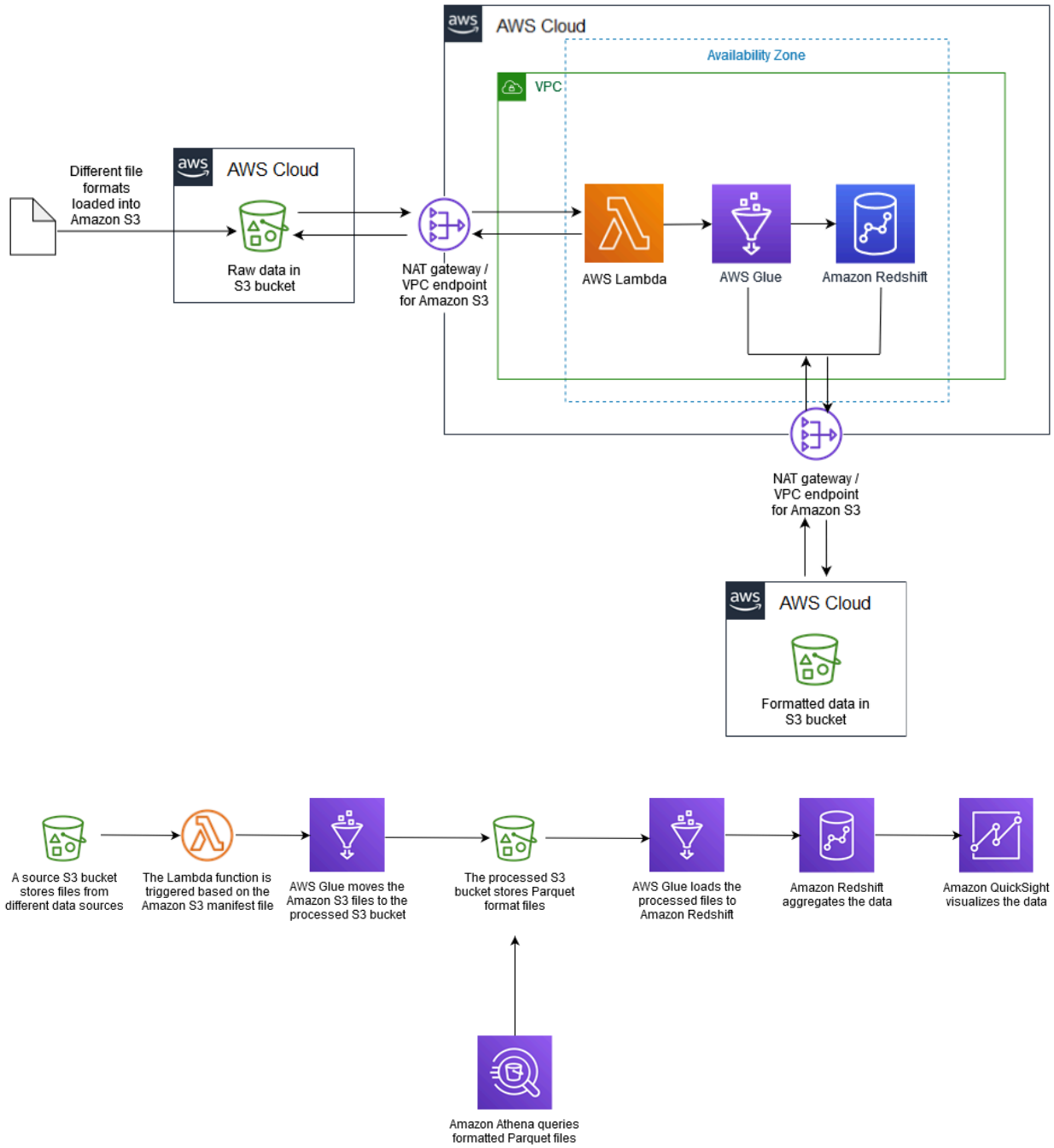
Build an ETL service pipeline to load data incrementally from Amazon S3 to Amazon Redshift using AWS Glue

While the AWS Free Tier offers limited free usage for various services, Amazon Redshift is not included in the Free Tier.The Redshift free trial is a separate offering designed to help new users explore the service without incurring charges.

We can load incremental data changes from Amazon S3 into Amazon Redshift by using AWS Glue, performing extract, transform, and load (ETL) operations.

The source files in Amazon S3 can have different formats, including comma-separated values (CSV), XML, and JSON files. You can use AWS Glue to convert the source files into a cost-optimized and performance-optimized format like Apache Parquet. You can query Parquet files directly from Amazon Athena and Amazon Redshift Spectrum. You

can also load Parquet files into Amazon Redshift, aggregate them, and share the aggregated data with consumers, or visualize the data by using Amazon QuickSight.



Amazon Athena is an interactive query service that makes it easy to analyze data that's stored in Amazon S3. Athena is serverless and integrated with AWS Glue, so it can

directly query the data that's cataloged using AWS Glue. Athena is elastically scaled to deliver interactive query performance.

Create the S3 buckets and folder structure

- Analyze source systems for data structure and attributes.
- Define the partition and access strategy.
- Create separate S3 buckets for each data source type and a separate S3 bucket per source for the processed (Parquet) data.

Create a data warehouse in Amazon Redshift

- Launch the Amazon Redshift cluster with the appropriate parameter groups and maintenance and backup strategy.
- Create and attach the IAM service role to the Amazon Redshift cluster.
- Create the database schema.
- Configure workload management.

Create a secret in Secrets Manager

- Create a new secret to store the Amazon Redshift sign-in credentials in Secrets Manager
- Create an IAM policy to restrict Secrets Manager access.

Configure AWS Glue

- In the AWS Glue Data Catalog, add a connection for Amazon Redshift.
- Create and attach an IAM service role for AWS Glue to access Secrets Manager, Amazon Redshift, and S3 buckets.
- Define the AWS Glue Data Catalog for the source.
- Create an AWS Glue job to process source data.
- Create an AWS Glue job to load data into Amazon Redshift.
- (Optional) Schedule AWS Glue jobs by using triggers as necessary.

Create a Lambda function

- Create and attach an IAM service-linked role for AWS Lambda to access S3 buckets and the AWS Glue job.
- Create a Lambda function to run the AWS Glue job based on the defined Amazon S3 event.
- Create an Amazon S3 PUT object event to detect object creation, and call the respective Lambda function.

Amazon Redshift best practices for designing tables

As you plan your database, certain key table design decisions heavily influence overall query performance. These design choices also have a significant effect on storage requirements, which in turn affects query performance by reducing the number of I/O operations and minimizing the memory required to process queries. Best practices for optimizing query performance.

- Choose the best sort key
- Choose the best distribution style
- Let COPY choose compression codings
- Define Primary Key and Foreign Key constraints
- Use the smallest possible column size
- Use date/time data types for date columns

Loading data in Amazon Redshift

One popular source of data to load are Amazon S3 files. Some of the methods to use with starting from an Amazon S3 source:

- COPY command
- COPY... CREATE JOB command (auto-copy)
- Load from data lake queries
- Streaming ingestion
- Running data lake queries
- Batch loading using Amazon Redshift query editor v2
- Load data from a local file using Amazon Redshift query editor v2

A COPY command is the most efficient way to load a table. You can also add data to your tables using INSERT commands, though it is much less efficient than using COPY. The COPY command is able to read from multiple data files or multiple data streams simultaneously. Amazon Redshift allocates the workload to the Amazon Redshift nodes and performs the load operations in parallel, including sorting the rows and distributing data across node slices.

Tutorial: Loading data from Amazon S3

https://docs.aws.amazon.com/redshift/latest/dg/tutorial-loading-data.html?utm_source=chatgpt.com

Work with semistructured data using Amazon Redshift SUPER

With the new SUPER data type and the PartiQL language, Amazon Redshift expands data warehouse capabilities to natively ingest, store, transform, and analyze semi-structured data. Semi-structured data (such as weblogs and sensor data) fall under the category of data that doesn't conform to a rigid schema expected in relational databases. It often contain complex values such as arrays and nested structures that are associated with serialization formats, such as JSON.

The schema of the JSON can evolve over time according to the business use case. Traditional SQL users who are experienced in handling structured data often find it challenging to deal with semi-structured data sources such as nested JSON documents due to lack of SQL support, the need to learn multiple complex functions, and the need to use third-party tools.

With the introduction of the SUPER data type, Amazon Redshift provides a rapid and flexible way to ingest JSON data and query it without the need to impose a schema. This means that you don't need to worry about the schema of the incoming document, and can load it directly into Amazon Redshift without any ETL to flatten the data. The SUPER data type is stored in an efficient binary encoded Amazon Redshift native format.

The SUPER data type can represent the following types of data:

- An Amazon Redshift scalar value:
    - A null
    - A Boolean
    - Amazon Redshift numbers, such as SMALLINT, INTEGER, BIGINT, DECIMAL, or floating point (such as FLOAT4 or FLOAT8)
    - Amazon Redshift string values, such as VARCHAR and CHAR
- Complex values:
    - An array of values, including scalar or complex
    - A structure, also known as *tuple* or *object*, that is a map of attribute names and values (scalar or complex)

After the semi-structured and nested data is loaded into the SUPER data type, you can run queries on it by using the PartiQL extension of SQL. PartiQL is backward-compatible to SQL. It enables seamless querying of semi-structured and

structured data and is used by multiple AWS services. With PartiQL, the query engine can work with schema-less SUPER data that originated in serialization formats, such as JSON. With the use of PartiQL, familiar SQL constructs seamlessly combine access to both the classic, tabular SQL data and the semi-structured data in SUPER. You can perform object and array navigation and also unnesting with simple and intuitive extensions to SQL semantics.

These four steps together form a complete pipeline for taking unstructured or semi-structured data from S3, processing it into structured form, and loading it efficiently into Amazon Redshift for analytics or reporting.