What?

Fully managed, petabyte-scale cloud data warehouse service. It also includes **Redshift Spectrum** that **runs SQL queries** directly against structured or unstructured data in Amazon **S3 without loading** them into Redshift cluster.

Why?

Redshift lets us run complex, analytic queries against **structured data and semi-structured data**, using sophisticated query optimization, **columnar storage** on high-performance storage like SSD, and massively parallel query execution.

Where?

AWS cloud based OLAP solution to store petabytes of information without owning infrastructure(Paas)

How?

We can connect via AWS CLI, AWS SDK, APIs, console

AWS Step Functions

What?

A **workflow**(or State Machine) **of steps**(or tasks) where the output of one step acts as an input to the next. Each step in an application executes in order, as defined by business logic.

Why?

With its built-in operational controls, Step Functions manages **sequencing, error handling, retry logic, state management, parameter passing** removing a significant operational burden from developers.

Where?

Mobile Apps management. Ex) Uber, Zomato, FoodPanda apps where right from order placement till order dispatch happens in sequence of steps. User's new request-> check User's details(for pending payment, authenticity)->check for nearby service riders->check with restaurant for food availability->Place an order and assign a rider->Close the order->Feedback

How?

We can connect via AWS CLI, AWS SDK, APIs, Console

What?

A **serverless data integration service** that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development.

Why?

Run Spark/Python code without managing Infrastructure at nominal cost. You pay only during run time of the job. Also pay storage cost for Data Catalog objects

Where?

Glue can automatically **discover** both **structured and semi-structured data** stored in your data lake on Amazon **S3**, data warehouse in Amazon **Redshift**, and various **databases** running on **AWS/on-premises** using JDBC. It provides a unified view of your data via the Glue Data Catalog that is available for ETL, querying and reporting using services like Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum.

How?

We can connect via AWS CLI, AWS SDK, APIs, Console

What?

Amazon Simple Notification Service is a notification service provided as part of Amazon Web Services since 2010. It provides a low-cost infrastructure for the mass delivery of messages, predominantly to mobile users

Why?

Out of the box solution with low operational overhead to deliver notifications via following endpoints

- HTTP/HTTPS
- Email/Email-JSON
- Amazon Kinesis Data Firehose
- Amazon SQS
- AWS Lambda
- Platform application endpoint
- SMS

Where?

In Alarming applications

How?

We can connect API/SDK/CLI calls, console

What?

Amazon Virtual Private Cloud (Amazon VPC) is a service that lets us launch AWS resources in a logically isolated virtual network we define. We have complete control over virtual networking environment, including selection of our own IP address range, creation of subnets, and configuration of route tables and network gateways.

Why?

There are no additional charges for creating and using the VPC itself. Usage charges for other Amazon Web Services, including Amazon EC2, Elastic IP address still apply at published rates for those resources, including data transfer charges

Where?

All environments in AWS cloud. If VPC is not created by user then he/she is bound to use the default VPC for most of the products

How?

We can connect via AWS CLI, AWS SDK, APIs, Console

What?

Amazon QuickSight is a scalable, serverless, embeddable, machine learning-powered business intelligence (BI) service built for the cloud.

Why?

The first BI service to offer pay-per-session pricing, where you only pay when your users access their dashboards or reports, making it cost-effective for large scale deployments. It can connect to wide variety of sources like Redshift, S3, Dynamo, RDS, files like JSON, text,csv, tsv, etc, jira, salesforce and on-premise oracle, sqlserver

Where?

A visualization Paas tool available in AWS. It supports wide variety of charts like bar, pie, donut, scatterplot, heatmap, treemap, etc. It also has Autograph where Quicksight will pick a chart category based on columns chosen and theirdata format

How?

We can control users access via IAM integration or adding users directly in Quicksight Console

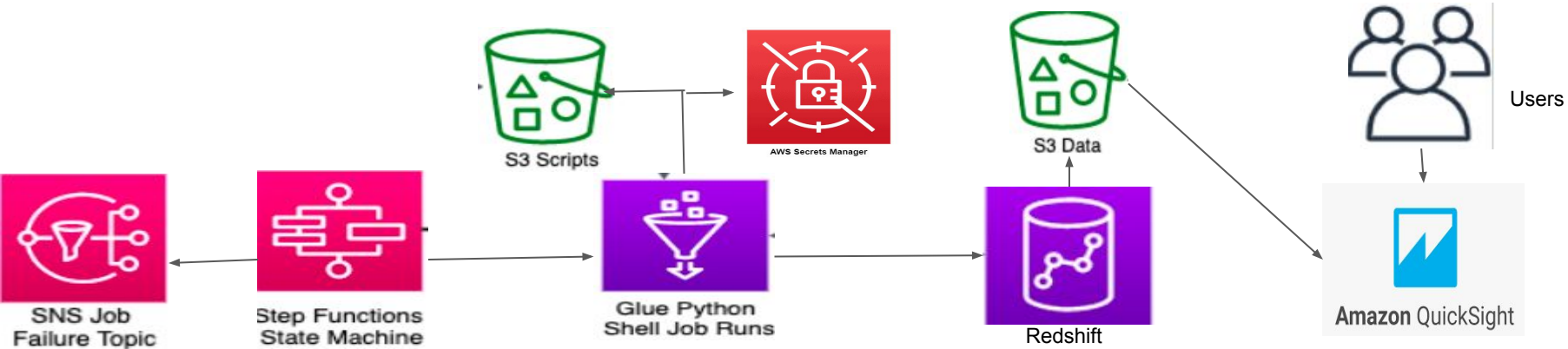
Orchestrate Redshift ETL workflow using Glue and Step Functions

Its very common to use Redshift as data-warehousing tool in AWS cloud. However, there are quite some ways to orchestrate the loading/unloading/querying Redshift. In this solution, we will use in-house AWS tools to orchestrate end-to-end loading and deriving business insights. Since it uses in-house tools, availability and durability of the solution is guaranteed by AWS

1. The state machine launches a series of runs of an AWS Glue Python Shell job with parameters for retrieving database connection information from AWS Secrets Manager and an .sql file from S3.
2. 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. **For Task 2:** The cluster executes an aggregation query and exports the results to another Amazon S3 location via UNLOAD.

3. The state machine may send a notification to an Amazon Simple Notification Service (SNS) topic in the case of pipeline failure.
4. Users can query the data from the cluster and/or retrieve report output files directly from S3/Redshift using Quicksight



VPC

1. Create a VPC “myProjectVPC” in us-east-1 region with IPV4 CIDR range: 10.71.0.0/16
2. Create the following subnets:
   a. Public Subnet A - 10.71.0.0/20
   b. Private Subnet A - 10.71.16.0/20
   c. Public Subnet B - 10.71.32.0/20
   d. Private Subnet B - 10.71.48.0/20
3. Create Internet Gateway-> “myprojectinternetgateway” and assign it to Public subnets A & B
4. Allocate a NAT Gateway-> “myprojectNAT” and allocate an Elastic IP address
5. Create 4 route tables, one for each subnet.
   a. 2 Public subnets(A&B) will have Internet Gateway referred if traffic is routed for 0.0.0.0/0
   b. 2 Private subnets(A&B) will have NAT Gateway referred if traffic is routed for 0.0.0.0/0

IAM

1. Create “myproject_gluerole” with AWSGlueServiceRole and a new policy using glue_demo_policy.json(in project resources)
2. Create “myproject_redshiftrole” with a new policy using redshift_demo_policy.json(in project resources)
3. Create “myproject_stepfunctionsrole” with a new policy using step_functions_demo_policy.json(in project resources)

Redshift

1. Create “myprojectcluster” as redshift cluster name after creating “Subnet Group” from Config tab. Subnet group should hold 2 private subnets(in 2 different regions) from VPC created. Choose default database name as “reviews” instead of “dev”
2. Make a note of admin/master user name and password
3. Click on Editor(on left below Clusters)->Connect to Database->Create New Connection->Store the secret in Secrets Manager->key in your cluster details->Give a name for the secret
4. Check the secrets manager for redshift secret. Use this secret as “db_creds” parameter in glue job

S3

1. Create a new S3 bucket “redshift-databucket<junk numbers> to hold the extract from redshift(this is the source for quicksight)
2. Create a new S3 bucket “redshift-scriptbucket<junk numbers> and create 2 folders: python, sql
3. Upload following files from project resources section in the same hierarchy:
    1) **python**
        a) rs_query.py
        b) redshift_module-0.1-py3.6.egg
    2) **sql**
        a) reviewsschema.sql
        b) topreviews.sql
        c) etl.sql

Glue

1. A connection in Glue Console pointing to “myprojectcluster” and test the connection using “myproject_gluerole” role. Check the policies if the connection fails**(step 1 in IAM)**
2. Create a new job using “Python Shell” option and choosing rs_query.py(as source file), *.egg file (as python dependency file) and use the following job parameters:

--file **sql/reviewsschema.sql (from step 2 in S3)**

--db_creds **redshiftqueryeditor-awsuser-myredshiftsecret (from step 4 in Redshift)**

--db **reviews (from step 1 in Redshift)**

--bucket **redshift-scriptbucket<junk numbers> (from step 2 in S3)**

3. Select the connection created in step 1 for connecting to Redshift cluster
4. Run the job manually
5. Check the logs for any issues. Ensure to re-check the parameters for any naming issue

SNS

1. Create a **Standard** Simple Notification Service(SNS) named “alarm-topic” and copy the arn from topic dashboard page
2. Create a subscription to SNS topic by adding your email-id and “EMAIL” as option
3. Accept subscription request by logging into your email account. Only then you will start receiving messages

Step Functions

4. Create a state machine and copy paste state_machine.json(in project resources) and save the definition using “Myproject_StepFunction” name
5. Update the definition with the parameter values similar to your succeeded Glue job except for “file” parameter
6. Update the SNS arn to your SNS topic arn **(Step 1 from SNS)**
7. Attach the “myproject_stepfunctionsrole” role to state machine **(step 3 from IAM)**
8. Click “Start Execution” button and check out the job status
9. Ensure Glue manual run succeeded before coming to this step. Then ensure policy and parameters in the state machine definition is correct as per your environment(Choose “Edit Definition” to alter the JSON file any time you want.

Quicksight

1. Activate Quicksight subscription under “Standard” edition
2. Choose the same region as your S3 account by clicking on right top corner Account name dropdown
3. Choose Manage Quicksight(right top corner Account name dropdown)-> Choose S3 bucket that you want to connect to
4. Click on “Create Dataset” and add “S3” and upload the manifest.json(in project resources by altering the name of the file according to your extract in redshift_databucket<junk numbers>
5. Rename columns to appropriate ones to avoid confusion as follows:

   Old Column: New Column

   Column 1: marketplace

   Column 2: product_category

   Column 3: product_title

   Column 4: review_id

   Column 5: helpful_votes

   Column 6: average_stars
6. Click on Analysis-> create a new one-> add “Add Visuals” on top left to add more charts and populate it by dragging and dropping corresponding column names