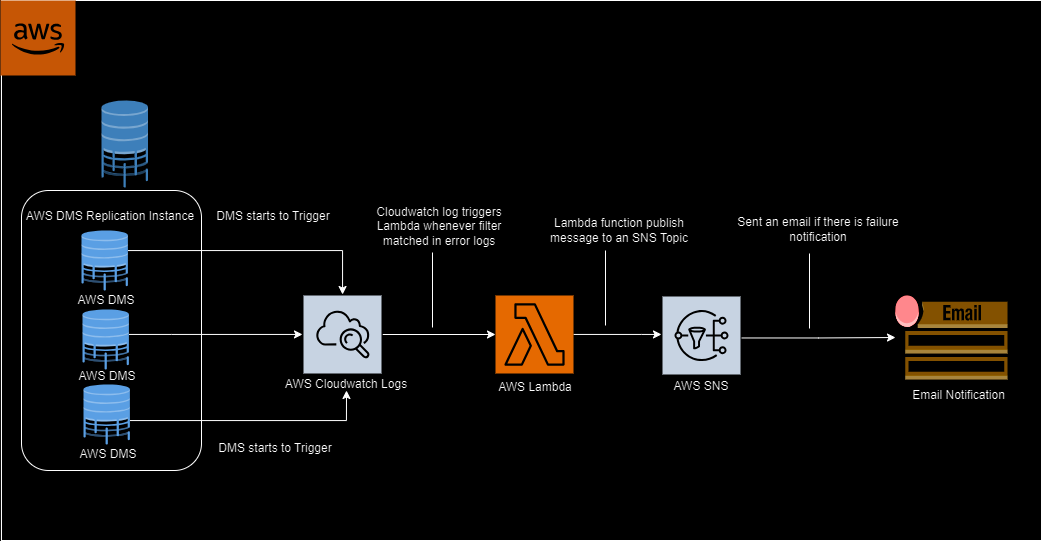
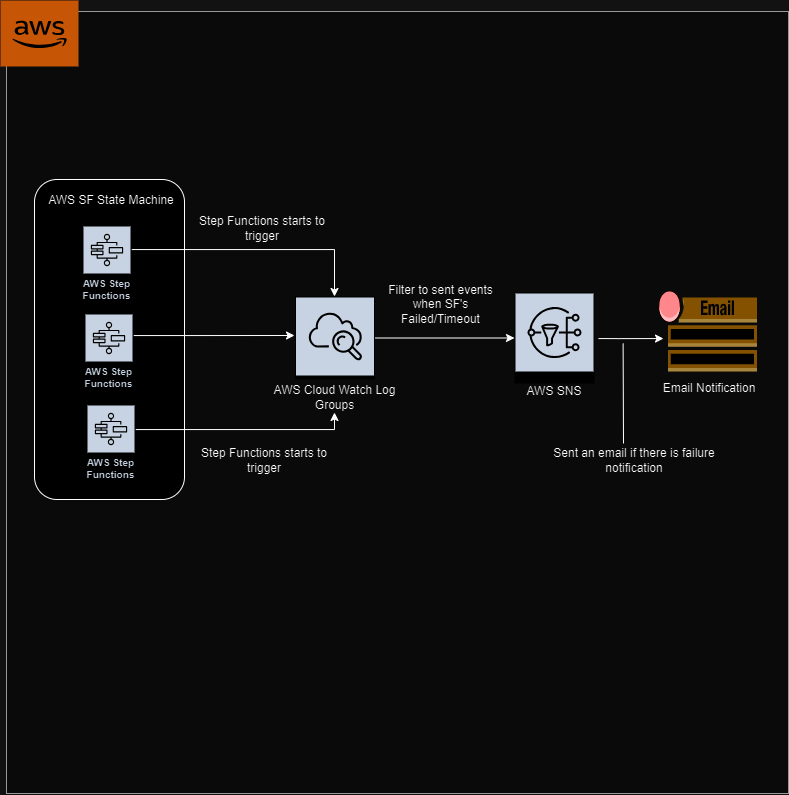
**Automation on Job Monitoring by creating an alert mechanism and Data Validation**

We can use AWS services to Automate Database Migration Services and corresponding Step Functions.

**AWS Services –** CloudWatch, SNS, Lambda and Cloud Formation Stack.

1. **Automate Monitoring and Alerting DMS task -**  
   Description of above solution diagram for DMS monitoring and alerting.  
   CloudWatch stores all the logs of DMS tasks meanwhile they are failed, success etc. Here we only use the Failed/Timeout event rule. CloudWatch will trigger Lambda for sending messages and publishing to SNS topic. In SNS topic we’ll provide the DL Email’s address. Whenever any SNS topic is trigger and Emails Subscriber will get the notifications of failure DMS.

**2. Automate Monitoring and Alerting Step Functions -**  


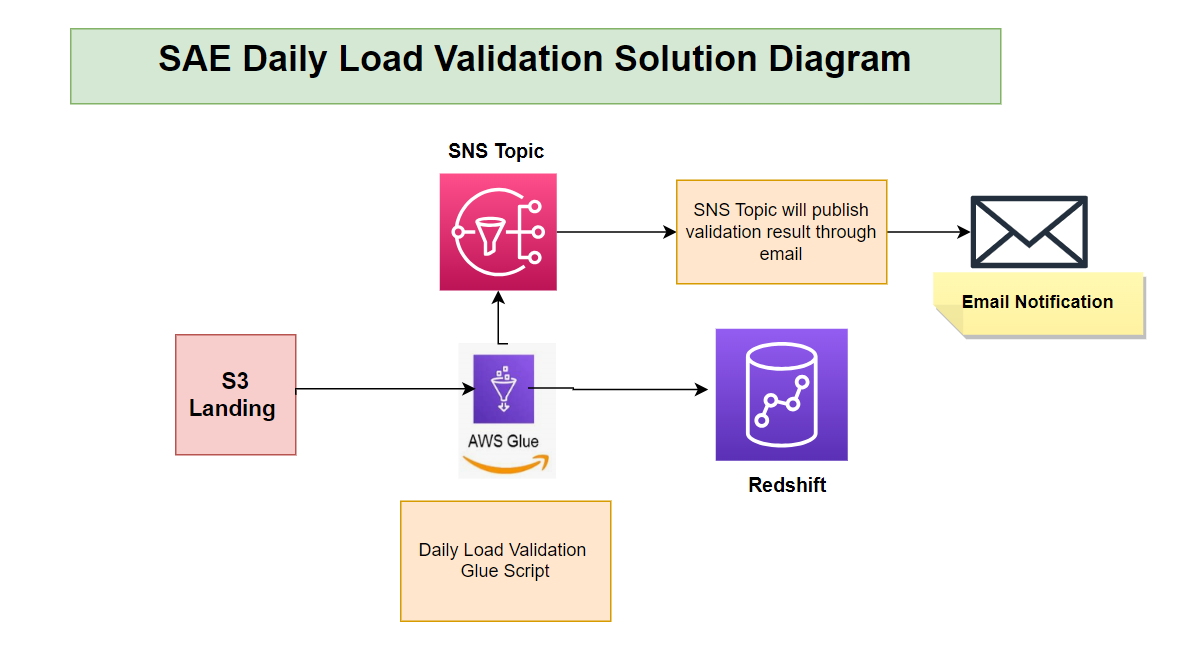
Description of solution diagram for Step Function for Monitoring and Alerting -  
  
If there is any Step Function will fail then event rule will be trigger and send notification to SNS Topic where we have DL Emails. After this Emails subscriber will get the Failure email.

**NOTE –** For **DMS** and **Step Functions** monitoring all the AWS resources are launched by AWS CloudFormation Stack.

**3. Automate Data Validation -**

Solution Diagram –

**AWS Services –** S3 bucket, SNS, AWS Glue and Redshift cluster.



**Introduction:**

This document outlines the validation procedures and results for the AWS Glue job responsible for integrating data between Amazon S3 and Amazon Redshift. The procedures detailed in this document specifically address the SAE daily load validation.

**Validation Steps**

1. **S3 Integration**
2. The script successfully reads Parquet files from the specified S3 path for each table.
3. Row counts for each S3 table are accurately calculated.
4. The script identifies and handles cases where the Parquet file is empty or doesn't exist.
5. **Redshift Integration**
6. The script successfully connects to Amazon Redshift using the specified connection options.
7. Row counts for each Redshift table are accurately obtained.
8. The script handles exceptions during the Redshift integration process and logs relevant information.
9. **Validation Result Comparison**
10. The script successfully joins and compares the row counts between S3 and Redshift for each table.
11. The validation results clearly indicate whether the row counts match or not.
12. Any differences in row counts are appropriately logged or reported.
13. **General Script Validation**
14. The script executes without errors and successfully commits the job.
15. Logging is implemented effectively, capturing relevant information for debugging and monitoring.
16. The script adheres to AWS Glue and Python best practices.
17. **Validation Results**

Daily load validation results are communicated via email through an SNS (Simple Notification Service) topic.

1. **Conclusion**

The AWS Glue job has been validated for S3 integration, Redshift integration, result comparison, and general script execution. Any identified issues have been documented, and the script is considered ready for deployment to the <Development/Production> environment.