Amazon Customer Reviews Analysis Using AWS

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose of this document 4

1.2 Intended Audience 4

1.3 Scope 4

1.4 Definitions and acronyms 4

1.4.1 Definitions 4

1.4.2 Acronyms and abbreviations 5

1.5 References 5

2. Background and Objectives 5

3. Organization 5

*3.1* Project group 5

3.2 Customer 6

4. Development process 6

*5.* Deliverables 6

6. Project risks 6

7. Communication 7

7.1 Canvas 7

7.2 Git 7

8. Project plan 7

8.1 Time schedule 7

8.1.1 Remarks 7

8.2 Test plan 9

8.2.1 Testing Remarks 10

9. References 10

# Introduction

## Purpose of this document

The purpose of this document is to provide a detailed project description of the application called Amazon Customer Reviews Analysis Using AWS, which is designed to help business analyst get detailed insights from customer reviews to help companies make better products and increase their sales. This document includes details about organization, roles, deliverables, project risks, time plans and financial plans.

## Intended Audience

This document shall be used in all phases of the project as a guideline. Intended audiences of this project are all project stakeholders:

* project supervisor- Siddharth Magidewar
* project leader- Jashwanth Kumar Enabothula.
* team members- Siddharth Magidewar, Jashwanth Kumar Enabothula.

## Scope

Our project aims to analyze the Amazon Customer Reviews data from two different data sources, one with a data length of 2 million reviews and a second data source of 1 million reviews with various graph representations to interpret these datasets.

## Definitions and acronyms

### Definitions

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
| Amazon Customer Reviews Analysis Using AWS | The name of the project |
| Project Supervisor | Siddharth Magidewar |
| Project Leader | Jashwanth Kumar Enabothula |
| Team Member | Siddharth Magidewar, Jashwanth Kumar Enabothula |
| Milestone | Nov-2021 |
| Git | Version control system that will be used in this project |
| Scrum | An iterative and incremental agile software development method for managing software projects and product or application development |
| Kunagi | Web-based tool for integrated agile project management and collaboration based on Scrum |
| Scrum sprint | Weekly |
| Scrum master | Siddharth Magidewar |
| Product owner | Jashwanth Kumar Enabothula |

### Acronyms and abbreviations

|  |  |
| --- | --- |
| **Acronym or**  **abbreviation** | **Definitions** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## References

1. <https://s3.amazonaws.com/amazon-reviews-pds/tsv/sample_us.tsv>
2. <https://s3.amazonaws.com/amazon-reviews-pds/tsv/sample_fr.tsv>

# Background and Objectives

**Abstract:**

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.ETL was developed to integrate and load data for calculation and analysis as databases become more popular in the 1970s. It eventually became the dominant way of processing data for data warehousing operations.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.

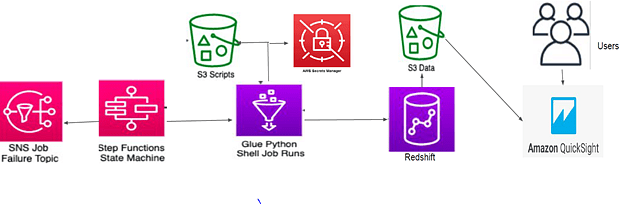
An organization's ETL is frequently used to Retrieve data from legacy systems,To improve data accuracy and reliability, clean the data.Update a target database with data.

It is prevalent to use Redshift as a data-warehousing tool in the AWS cloud. However, there are quite some ways to orchestrate the loading, unloading and querying Redshift. 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

**Objective**

Our project aims to analyze the Amazon Customer Reviews data from two different data sources, one with a data length of 2 million reviews and a second data source of 1 million reviews with various graph representations to interpret these datasets.

# Architecture & High-Level Design



.

# Organization

## Project group

|  |  |  |
| --- | --- | --- |
| **Name** | **Initials** | **Responsibility (roles)** |
| Siddharth Magidewar | SM | visualization, documentation |
| Jashwanth Kumar Enabothula | EJK | Data sourcing/modeling and Development |

## Customer

The target customers are listed below:

* 1 Business Analyst
* 2 Customers

# Development process

1. The state machine launches a succession of AWS Glue Python Shell jobs, each with parameters for retrieving database connection information from AWS Secrets Manager and a SQL file from S3.

2. Each execution of the AWS Glue Python Shell task uses the database connection information to connect to the Amazon Redshift cluster and submit the queries in the SQL file.

Task 1: The cluster reads data from S3 and loads it into an Amazon Redshift table using Amazon Redshift Spectrum.

Task 2: The cluster runs an aggregation query and then uses UNLOAD to export the results to another Amazon S3 location.

3. In the event of a failure, the state machine may send a notification to an Amazon Simple Notification Service (SNS) topic.

4. Users can query he data from the cluster or retrieve report output files directly from S3/Redshift using QuickSight.

# Deliverables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **To** | **Output** | **Planned week** | **Promised week** | **Late +/-** | **Delivered week** | **Notes** |
| **VPC** | **Created Secure connections between s3 bucket and Redshift** | **Sept 3rd Week** | **Sept 3rd Week** | **No** | **Sept 3rd Week** |  |
| **Redshift** | **Created Redshift Cluster** | **Oct 1st Week** | **Oct 1st Week** | **No** | **Oct 1st Week** |  |
| **Glue Jobs** | **Created Glue Jobs** | **Oct 2nd Week** | **Oct 2nd Week** | **No** | **Oct 2nd Week** |  |
| **S3** | **Created S3 Bucket** | **Oct 3rd Week** | **Oct 3rd Week** | **No** | **Oct 3rd Week** |  |
| **Querying and Data Transformation** | **Transformed data** | **Oct 4th Week** | **Oct 4th Week** | **No** | **Oct 4th Week** |  |
| **SNS and Step Functions** | **Created Simple Notification System and Step Functions** | **November 1st Week** | **November 1st Week** | **No** | **November 1st Week** |  |
| **Visualization and Analytics** | **Data was analyzed using AWS QuickSight** | **November 2nd Week** | **November 2nd Week** |  | **November 2nd Week** |  |

# Project risks

|  |  |  |
| --- | --- | --- |
| **Possibility** | **Risk** | **Preventive action** |
| Cost risk for using S3 and RedShift Cluster on AWS | It could increase the overall cost of the project | Deleted both cluster and S3 bucket after obtaining desired output |
| Error in workflow | It could fail during the steps of execution | We have built workflow in Step Function to notify the error immediately in the form of email or SMS to the developer |

# Communication

The Team connected through Zoom weekly

## Collaboration

## Git

All source code and finished documentation will be uploaded to Github repository. ..

Repository URL: https://github.com/siddharth414/sjsu-data228

# Project plan

## Time schedule

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Milestone**  **Description** | **Responsible Dept./Initials** | **Finished week**  **Plan** | **Forecast Week** | **+/-** | **Actual** | **Metr.** | **Rem.** |  |
|  | **VPC** | **Siddharth** | **Sept 3rd Week** | **Sept 3rd Week** | **No** | **Sept 3rd Week** |  |  |  |
|  | **Redshift** | **Jashwanth** | **Oct 1st Week** | **Oct 1st Week** | **No** | **Oct 1st Week** |  |  |  |
|  | **Glue Jobs** | **Jashwanth** | **Oct 2nd Week** | **Oct 2nd Week** | **No** | **Oct 2nd Week** |  |  |  |
|  | **S3** | **Siddharth** | **Oct 3rd Week** | **Oct 3rd Week** | **No** | **Oct 3rd Week** |  |  |  |
|  | **Querying and Data Transformation** | **Jashwanth** | **Oct 4th Week** | **Oct 4th Week** | **No** | **Oct 4th Week** |  |  |  |
|  | **SNS and Step Functions** | **Jashwanth** | **November 1st Week** | **November 1st Week** | **No** | **November 1st Week** |  |  |  |
|  | **Visualization and Analytics** | **Siddharth** | **November 2nd Week** | **November 2nd Week** | **No** | **November 2nd Week** |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

### Remarks

|  |  |
| --- | --- |
| **Remark Id** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |

## Test plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | 001 | Phase: | 1 | Author: | Jashwanth | Date: Nov 2021 |
| Test Category: | | **State Machine Execution** | | |  |  |
| Software Product: | | AWS Step Function | | | |  |
| Test Title: | | Glue Job Execution | | | | |
| Test Purpose: | | To check if all jobs are executed | | | | |
| Test Setup: | | State machines were created using JSON templates having task definitions and workflow logic | | | | |
| Prerequisites: | | AWS Glue Python Shell job | | | | |
| Procedure: | | Navigate to Step Functions in the AWS Console and look for a state machine with a given name .Now navigate to state machine main page and select Start an Execution. | | | | |
| Checks: | | Check if we have selected correct state machine name | | | | |
| Expected Results: | | Successful job runs and data loading into S3 bucket | | | | |
| Result: | | Fail | | | | |
| Reason for Failure: | | Destination bucket name was incorrect | | | | |
| Remarks: | | Bucket name has to be corrected | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | 002 | Phase: | 1 | Author: | <username> | Date: |
| Test Category: | | **State Machine Execution** | | |  |  |
| Software Product: | | AWS Step Function | | | |  |
| Test Title: | | Glue Job Execution | | | | |
| Test Purpose: | | To check if all jobs are excuted | | | | |
| Test Setup: | | State machines were created using JSON templates having task definitions and workflow logic | | | | |
| Prerequisites: | | AWS Glue Python Shell job | | | | |
| Procedure: | | Navigate to Step Functions in the AWS Console and look for a state machine with a given name .Now navigate to state machine main page and select Start an Execution. | | | | |
| Checks: | | Check if we have selected correct state machine name | | | | |
| Expected Results: | | Successful job runs and data loading into S3 bucket | | | | |
| Result: | | Pass | | | | |
| Reason for Failure: | |  | | | | |
| Remarks: | |  | | | | |

### Testing Remarks

|  |  |
| --- | --- |
| **Remark Id** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |

# References

* [1.https://s3.amazonaws.com/amazon-reviews-pds/tsv/sample\_us.tsv](file:///C:\Users\Checkout\Downloads\1.%09https:\s3.amazonaws.com\amazon-reviews-pds\tsv\sample_us.tsv)
* <https://aws.amazon.com/blogs/big-data/orchestrate-amazon-redshift-based-etl-workflows-with-aws-step-functions-and-aws-glue/>
* <https://docs.aws.amazon.com/step-functions/latest/dg/sample-etl-orchestration.html>
* <https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/orchestrate-an-etl-pipeline-with-validation-transformation-and-partitioning-using-aws-step-functions.html>