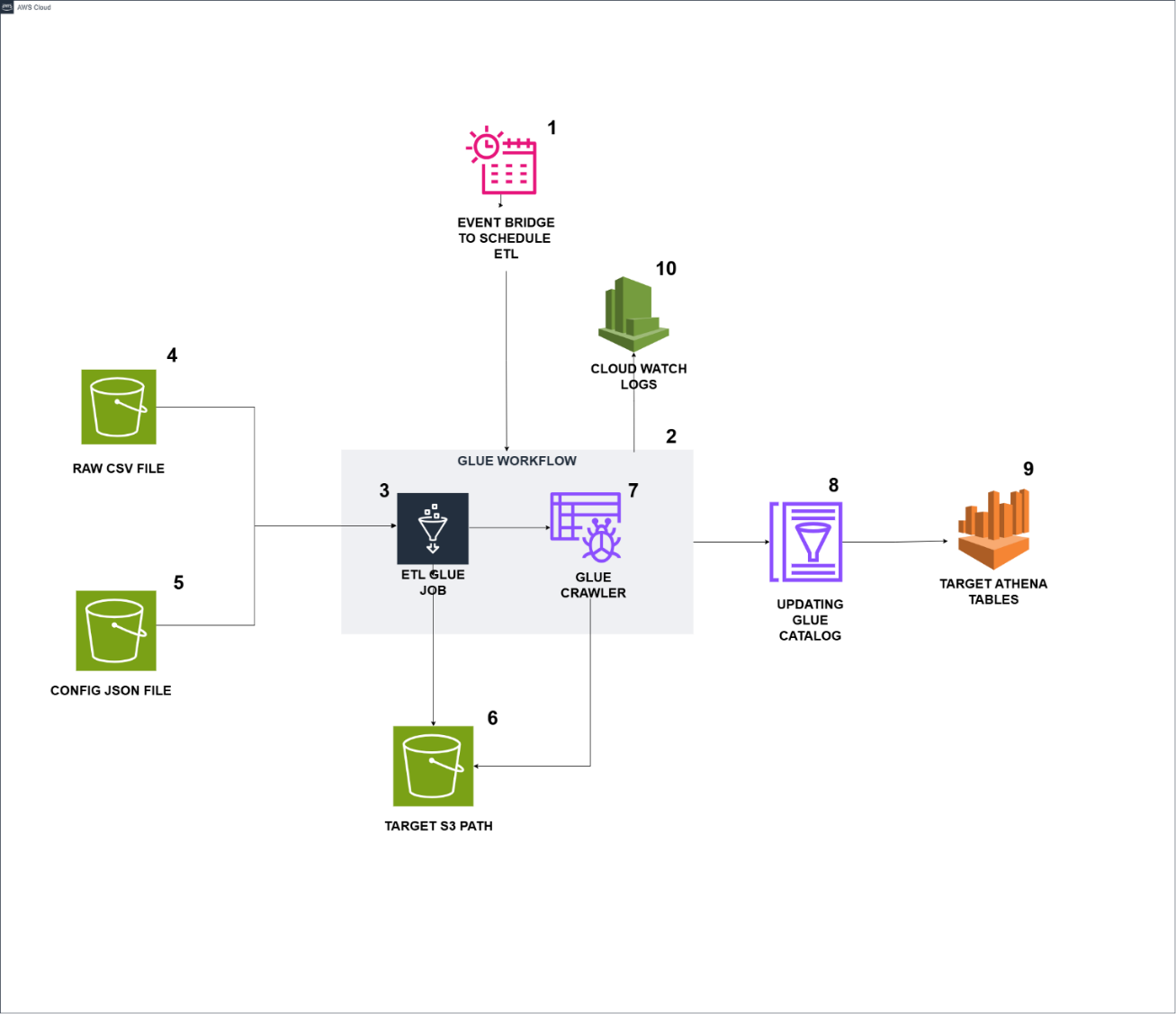
**Health-Score & Containment Project Help Document**

**ARICHITECTURE DIAGRAM FOR HEALTHSCORE ETL:**

****

**DESCRIPTION:**

1. Event Bridge triggers the Glue workflow based on the scheduled time.
2. The Glue workflow is triggered by Event Bridge and initiates the run.
3. A Glue job starts running within the Glue workflow using an On-Demand trigger.
4. The Glue job reads raw csv file data from the s3 path.
5. The Glue job reads config json file data from the s3 path.
6. The Glue job start process the data and apply all the calculation logics then it writes the data into target s3 path for these 4 tables (healthscore\_master, healthscore\_actual\_data, healthscore\_percent\_both, healthscore\_calculations).
7. Upon successful completion of the Glue job, the Glue crawler is triggered and starts running.
8. The Glue Crawler will update the glue data catalog with proper date type formats.
9. The target tables are available in Athena, where the data can be queried.
10. Cloud Watch stores logs for both the Glue jobs and the Glue crawler.

**TABLE STRUCTURE : HEALTH SCORE**

|  |  |  |
| --- | --- | --- |
| **Table Name – healthscore\_master** | | |
| **Field Name** | **Data type** | **Comment** |
| metric\_id | Int | Auto generated number |
| metrics | string | From raw csv file |
| display\_names | string | Generated using json file |
| operating\_system\_type | string | From raw csv file |
| feature\_names | string | Generated using json file |
| unique\_identifier | string | Fetched from feature\_names |
| metric\_nature | string | Generated using json file |
| level\_1 | string | Generated using json file |
| level\_2 | string | Generated using json file |
| level\_3 | string | Generated using json file |
| level\_4 | string | Generated using json file |
| level\_5 | string | Generated using json file |
| hierarchy\_id | string | Generated using level 1,2,3,4,5 |
| parent\_id\_old | string | Generated using level 1,2,3,4,5 |
| metric\_sequence\_number | integer | Auto generated number |
| metric\_seqno | integer | Generated using json file |
| level\_no | integer | Generated using level 1,2,3,4,5 |
| feature\_id | integer | Auto generated number |
| feature\_seq\_num | integer | Generated using json file |
| parent\_id | integer | Generated using level 1,2,3,4,5 |
| create\_dt | date | Last ran date of the glue job |

|  |  |  |
| --- | --- | --- |
| **Table Name – healthscore\_actual\_data** | | |
| **Field Name** | **Data type** | **Comment** |
| metric\_id | integer | Fetched from **healthscore\_master table** |
| create\_dt | date | Last ran date of the glue job |
| Date | date | Pivoted all the Date columns from Raw file into rows |
| Value | integer | Pivoted all the integer values of Date columns from Raw file into rows |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name – healthscore\_percent\_both** | | | |
| **Field Name** | **Data type** | | **Comment** |
| metric\_id | integer | Fetched from **healthscore\_master table** | |
| create\_dt | date | Last ran date of the glue job | |
| feature\_name | string | Fetched from **healthscore\_master table** | |
| metrics | string | Fetched from **healthscore\_master table** | |
| display\_names | string | Fetched from **healthscore\_master table** | |
| operating\_system\_type | string | Fetched from **healthscore\_master table** | |
| Date | Date | Pivoted all the Date columns from Raw file into rows | |
| Value | float | Calculated % values for each Date column  Calculated Both rows for each metrics Pivoted all the calculated data of Date columns into rows | |

|  |  |  |
| --- | --- | --- |
| **Table Name – healthscore\_calculations** | | |
| **Field Name** | **Data type** | **Comment** |
| metric\_id | integer | Fetched from **healthscore\_master table** |
| create\_dt | date | Last ran date of the glue job |
| feature\_name | string | Fetched from **healthscore\_master table** |
| metrics | string | Fetched from **healthscore\_master table** |
| display\_names | string | Fetched from **healthscore\_master table** |
| operating\_system\_type | string | Fetched from **healthscore\_master table** |
| 1st quartile | double | Calculated using Python code |
| 3rd quartile | double | Calculated using Python code |
| iqr | double | Calculated using Python code |
| upper | double | Calculated using Python code |
| lower | double | Calculated using Python code |
| yesterday | double | Calculated using Python code |
| Last\_7\_days | double | Calculated using Python code |
| Last\_30\_days | double | Calculated using Python code |
| % change last 7 days | double | Calculated using Python code |
| % change last 30 days | double | Calculated using Python code |
| metric\_nature\_indicator | string | Calculated using Python code |
| colour\_indicator | string | Calculated using Python code |

**TABLE STRUCTURE : CONTAINMENT METRIC**

|  |  |  |
| --- | --- | --- |
| **Table Name – containement\_metric\_master** | | |
| **Field Name** | **Data type** | **Comment** |
| id | Int | Auto generated number |
| hs\_feature\_name | string | Generated using json file |
| hs\_feature\_id | string | Generated using json file |
| primary\_intent | string | Generated using json file |
| primary\_intent\_detail | string | Generated using json file |
| cont\_display\_metric\_name | string | Generated using json file |
| Cont\_display\_metric\_seq | string | Generated using json file |
| create\_dt | date | Last ran date of the glue job |

|  |  |  |
| --- | --- | --- |
| **Table Name – containment\_metric\_raw\_data** | | |
| **Field Name** | **Data type** | **Comment** |
| primary\_intent | string | Fetched form Omni table |
| primary\_intent\_detail | string | Fetched form Omni table |
| sub\_contact\_id | date | Fetched form Omni table |
| contact\_id | integer | Fetched form Omni table |
| selfservice\_containment | integer | Fetched form Omni table |
| intial\_channel | integer | Fetched form Omni table |
| lob | string | Fetched form Omni table |
| create\_dt | date | Fetched form Omni table |
| is\_active | boolean | Fetched form Omni table |
| contact\_dt | string | Fetched form Omni table |

|  |  |  |  |
| --- | --- | --- | --- |
| **able Name – containment\_metric\_data** | | | |
| **Field Name** | **Data type** | | **Comment** |
| sub\_contact\_id | integer | Fetched form **containment\_metric\_raw\_data** | |
| selfservice\_containment | date | Fetched form **containment\_metric\_raw\_data** | |
| intial\_channel | string | Fetched form **containment\_metric\_raw\_data** | |
| lob | string | Fetched form **containment\_metric\_raw\_data** | |
| contact\_dt | string | Fetched form **containment\_metric\_raw\_data** | |
| containment\_rate | string | Fetched form **containment\_metric\_raw\_data** | |
| hs\_feature\_name | Date | Fetched from **containement\_metric\_master** | |
| cont\_display\_metric\_name | float | Fetched from **containement\_metric\_master** | |
|  |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name – containment\_metric\_calculation** | | | |
| **Field Name** | **Data type** | | **Comment** |
| hs\_feature\_name | integer | Fetched from **containement\_metric\_master** | |
| Cont\_display\_metric\_name | date | Fetched from **containement\_metric\_master** | |
| yesterday | string | Calculated using Python code | |
| Last\_7\_days | string | Calculated using Python code | |
| Last\_30\_days | string | Calculated using Python code | |
| % Change Last 7 days | string | Calculated using Python code | |
| % Change Last 30 days | Date | Calculated using Python code | |
| create\_dt | float | Last ran date of the glue job | |

**GLUE JOB FUNCTIONALITIES**

* **Read Cleaned Raw Input Data:** Load the cleansed raw input CSV file for health score data.
* **Read Configuration File:** Load the configuration JSON file containing necessary mappings and settings.
* Generate Required Columns: Using the configuration file, generate essential columns such as metric ID, display names, feature name, unique identifiers, metric nature, hierarchical levels, sequence numbers, and creation date.
* **Load Master Data:** Store the generated columns in the appropriate data store, where a data crawler creates the healthscore\_master table.
* **Pivot Date Columns:** Transform the date columns from the cleansed raw input data into a row-based format, storing the values as a "Value" column for each metric.
* **Load Actual Data:** Store the pivoted data into the data store, where a data crawler creates the healthscore\_actual\_data table.
* **Metric Calculations:** Based on metric\_id and parent\_id, calculate the numerators, denominators, percentages, and both-row calculations for each metric.
* **Pivot Calculated Data:** Pivot the calculated percentage and both-row data from column-based format into row-based format, with metric details and values.
* **Load Percent and Both Data:** Store the pivoted percentage and both-row calculation data, allowing a data crawler to create the healthscore\_percent\_both table.
* **Statistical Calculations:** Calculate first and third quartiles, IQR (Interquartile Range), upper and lower bounds, as well as last 7-day and 30-day averages. Derive metric nature and color indicators based on these values.

**MONITORING THE JOB & RESULTS PATH**

* Check the CloudWatch logs (Glue output logs) after starting the glue job
* Make sure the new data is loaded in the below target s3 paths after successful completion of job

**healthscore\_master**: s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_master/

**healthscore\_actual\_data**: s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_actual\_data/

**healthscore\_percent\_both**: s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_percent\_both/

**healthscore\_calculations**: s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_calculations/

**PREREQUISITES TO RUN GLUE ETL:**

* CSV File S3 Path: s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_raw\_file/HS\_DataClean\_Nov.csv
* JSON File S3 Path: s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_json/healthscore\_dashboard.json
* Make sure the CSV raw file & JSON file are present in the S3 Path
* CSV Raw file should only have Metrics, Operating System Type, any given Months data
* JSON file should only have Feature Names, Feature Sequence Number, Display Names, Metric Nature, Metric Sequence Number, Level 1,2,3,4,5 metrics.

**CHANGES REQUIRED WHEN NEW MONTHS DATA ADDED IN CSV:**

* The new data should be added as a new file name in this s3 path

s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_raw\_file/

* The file structure of CSV should only have these columns Metrics, Operating System Type, and new month’s data
* Make sure the Job Bookmark is enabled in glue job before running

**CHANGES REQUIRED WHEN NEW METRICS ADDED IN CSV:**

* The file structure of CSV should only have these columns Metrics, Operating System Type, and new month’s data
* Make sure the Job Bookmark is enabled in glue job before running
* Update the new Metrics information in the JSON file (Feature Names, Feature Sequence Number, Display Names, Metric Nature, Metric Sequence Number, Level 1,2,3,4,5 metrics) before run the glue job

**STEPS TO RUN THE JOB AUTOMATICALLY:**

* Setup the scheduled time in Event Bridge
* Based on the scheduled time, glue workflow will automatically trigger and it will start run the glue job and then crawler
* The target athena tables will be automatically populated by crawler.

**STEPS TO RUN THE JOB MANUALLY:**

* Go to AWS Glue services 🡪 ETL Jobs 🡪 Healthscore\_dashboard
* Manually Run the job Healthscore\_dashboard job
* Monitor the glue job while running, by clicking the Runs 🡪 Run Details 🡪 Output Logs
* After successful completion of the job, check these output paths
* **healthscore\_master** : s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_master/
* **healthscore\_actual\_data** : s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_actual\_data/
* **healthscore\_percent\_both** : s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_percent\_both/
* **healthscore\_calculations** : s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_calculations/
* If the parquet file loaded in these paths, go to crawler section
* Manually run the Healthscore\_dashboard crawler
* After successful completion of the crawler, check these Athena tables (**healthscore\_master** , **healthscore\_actual\_data** , **healthscore\_percent\_both, healthscore\_calculations**) in **healthscrore database**