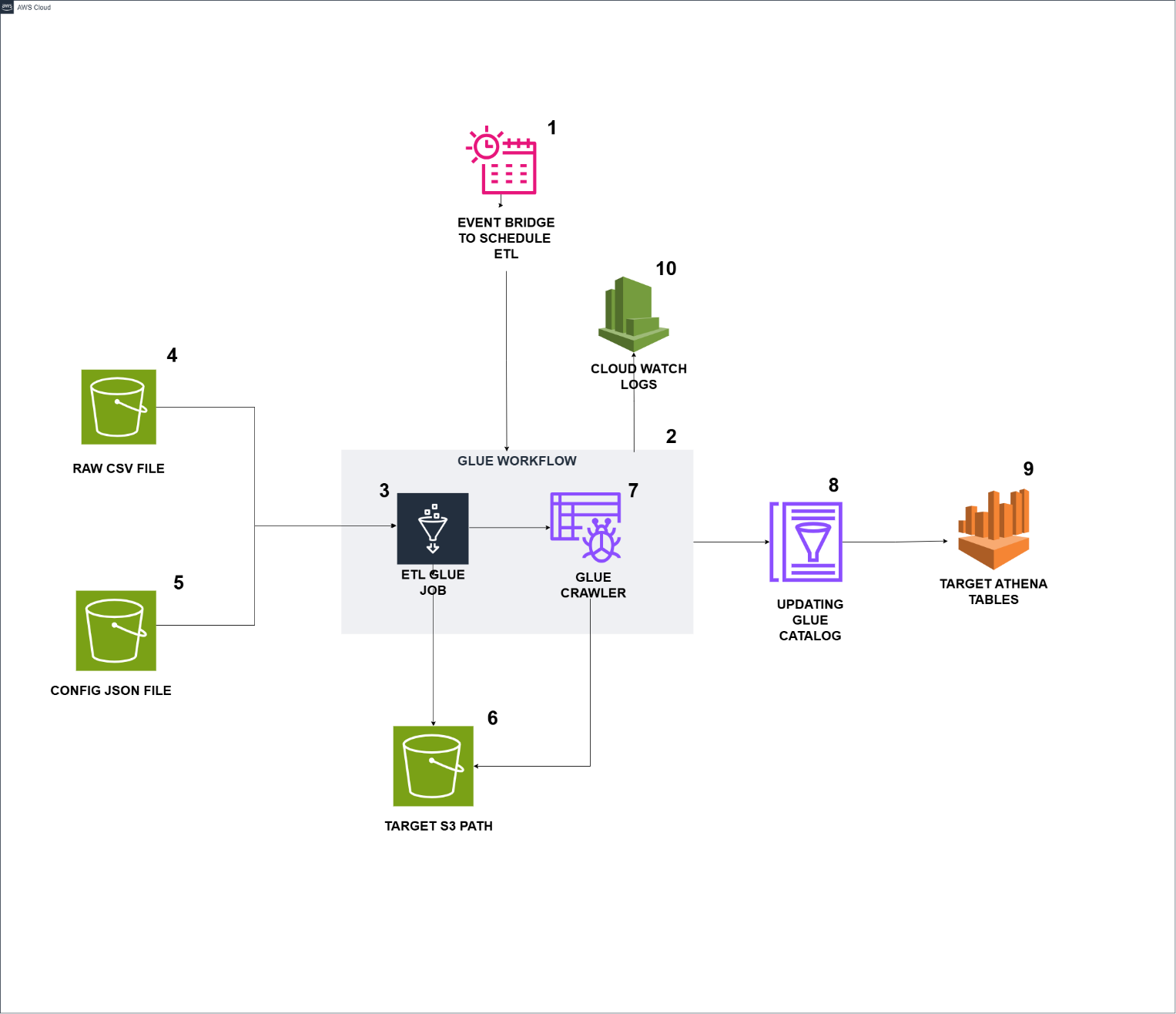
**HEALTHSCORE DASHBOARD**

**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:**

|  |  |  |
| --- | --- | --- |
| **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 |

**GLUE JOB CONFIGURATION :**

Job Name : healthscore\_dashboard

Job Type: Spark

Glue Version: Glue 5.0

Language: Python 3

Worker Type: G.1X

Maximum No. of Workers: 5

Job Bookmark: Enable

Job metrics: Enable

Job observability metrics: Enable

Continuous logging: Enable

Maximum Concurrency: 1

**IAM ROLES & PERMISSIONS:**

**IAM ROLE:** **cci-svc-glue-execution**

* S3 read & write permission required
* Cloud watch permission required to check the logs

**GLUE JOB RUNTIME**: 3 Minutes

**GLUE JOB FUNCTIONALITIES:**

* Reading cleansed raw input csv file from this s3 path
  + s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_raw\_file/HS\_DataClean\_Nov.csv
* Reading config json file from this s3 path
  + s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_json/healthscore\_dashboard.json
* Generating these columns with the help of JSON file
  + ( metric\_id, display\_names, feature\_name, unique\_identifier, metric\_nature, level1,level2,level3,level4, hierarchy\_id, parent\_id\_old, metric\_sequence\_num, metric\_seqno, levelno, feature\_id, feature\_seq\_num, parent\_id, create\_dt )
* These columns data are loaded into s3 path, then using crawler healthscore\_master table will be created
  + s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_master/
* From the cleansed raw input, pivoting the Date columns into row format with the actual numbers stored in Values column as row format.
  + (metric\_id, create\_dt, date, Value)
* These columns data are loaded into s3 path, then using crawler healthscore\_actual\_data table will be created
  + s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_actual\_data/
* Based on the metric\_id & parent\_id**,** numerators & denominators can be determined and the percentage calculation of each metrics is calculated.
* Based on the metric\_id & parent\_id**,** numerators & denominators can be determined and the Both rows calculation of each metrics is calculated.
* Then Percentage data & Both rows data will be pivoted, the Date columns into row format and the calculated percentage & Both rows data into row format as Value column
  + (metric\_id, create\_dt, feature\_name, metrics, display names, operating system type, date, Value)
* These columns data are loaded into s3 path, then using crawler healthscore\_percent\_both table will be created
  + s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_percent\_both/
* First Quartile, 3rd Quartile has been calculated using quantile function
* IQR has been calculated, difference of 3rd Quartile and 1st Quartile
* Based on parent\_id values & First Quartile, 3rd Quartile, IQR. the Upper & Lower column has been calculated
* The Last 7 & 30 days has been calculated with mean function
* The metric\_nature\_indicator & colour indicator has been calculated based on upper, lower columns
  + ( metric\_id, create\_dt, feature\_name, metrics, display name, operating system type, 1st quartile, 3rd quartile, iqr, upper, lower, yesterday, last 7 days, last 30 days, % change last 7 days, % change last 30 days, metric\_nature\_indicator, colour\_indicator)
* These columns data are loaded into s3 path, then using crawler healthscore\_calcuations table will be created
  + s3://cci-dig-aicoe-data-sb/processed/healthscore/healthscore\_calculations/

**MONITORING THE JOB & RESULTS PATH**

* Check the CloudWatch logs (Glue output logs) after started 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 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 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, new months data
* Make sure the Job Bookmark is enabled in glue job before run

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

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