**GCP Cost Dashboard – BETA Program**

This guide explores the [newly announced](https://aws.amazon.com/about-aws/whats-new/2023/10/aws-glue-apache-spark-connectivity-google-bigquery/) managed connector for Google BigQuery and demonstrates how to build a modern ETL pipeline for GCP Cost with AWS Glue Studio without writing code.

**Prerequisites**

1. Create a GCP Billing Export:
   1. One export is related to one billing account id in GCP.
   2. https://cloud.google.com/billing/docs/how-to/export-data-bigquery-setup
2. Create a GCP Pricing Export
   1. When enabling Cloud Billing export, **ensure to include Pricing data**.
   2. https://cloud.google.com/billing/docs/how-to/export-data-bigquery-setup#enable-bq-export
3. Configure BigQuery Connection Secret:
   1. To connect to Google BigQuery from AWS Glue, see and complete points 1-4 in [Configuring BigQuery connections](https://docs.aws.amazon.com/glue/latest/dg/aws-glue-programming-etl-connect-bigquery-home.html#aws-glue-programming-etl-connect-bigquery-configure). You must create and store your Google Cloud Platform credentials in a Secrets Manager secret, then associate that secret with a Google BigQuery AWS Glue connection.
      1. When selecting GCP IAM roles for your service account, add or create a role that would grant appropriate permissions to run BigQuery jobs to both read and write, and create BigQuery tables in the single billing dataset.
   2. Video to create the secret: 1:54-2:38 https://youtu.be/n6fkX5LpEYY?t=114

At the end of prerequisites, you should have the following values:

1. 1 or more BigQuery billing table
2. 1 or more BigQuery pricing
3. 1x Secret in AWS with json key of GCP SA stored in base64 with *credentials* as key

**Infrastructure Deployment Guide**

1. Open Cloudformation in your aws console, in your CID account.  
   <https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringText=&filteringStatus=active&viewNested=true>
2. Select ‘Stack’, ‘Create Stack’, ‘Upload a template file’.
3. Upload cid-gcp-costs.yaml
4. Insert parameters:
   1. Prefix => a prefix for the resources created
   2. BucketName => a bucket to store exctracted data. Provide a name based on your naming convention.
   3. ScriptBucketName => a bucket to store glue job code. Provide a name based on your naming convention.
   4. GCPFullTableName => this is a COMMADELIMITEDLIST type, and should includes your billing export tables ID delimited by a “,”.   
      example: value1,value2.  
      to retrieve your table ID, just select the three dots on the right of a table and click “**Copy ID**”.
   5. GCPPricingFullTableName => your pricing table id.   
      to retrieve your table ID, just select the three dots on the right of a table and click “**Copy ID**”.
   6. GCPConnectionName => a name for the connection. Default can be good or adapt based on your naming convention needs.
   7. GCPBillingLocation, GCPPricingLocation, GCPJobBookmarksKeys and TargetCatalogDBName are reserved and must not be changed.
5. Click on next until completing the form.
6. Launch the cloudformation and wait until it’s complete.
7. Test the import:
   1. Go on your newly created Glue Jobs and run them to start an import.
   2. Wait until completes.

**Dashboards Deployment Guide**

1. On cloudshell install cid with command: pip3 install -U cid-cmd
2. Upload GCP-Cost-Dashboard-2.yaml to your cloudshell.
3. Launch command: cid-cmd deploy --resources GCP-Cost-Dashboard-2.yaml