**BigQuery Health Check Metadata Upload Guide**



| Product Version | 4.8.2 |
| --- | --- |
| Document Type | Health Check Preparation Guide |
| Authors | BigQuery Data source Team |
| Reviewer | Red Team & Architects |
| Approver | CTO |
| Total Pages | 6 |
| Document Status | Draft |

Table Of Contents

[**1.1**](#_30j0zll) **Objectives 2**

[**1.2**](#_1fob9te) **Architecture 2**

[**1.3**](#_3znysh7) **Prerequisite 2**

[**1.4**](#_2et92p0) **Upload BigQuery Metadata for health check. 3**

[**1.5**](#_tyjcwt) **Output Files 5**

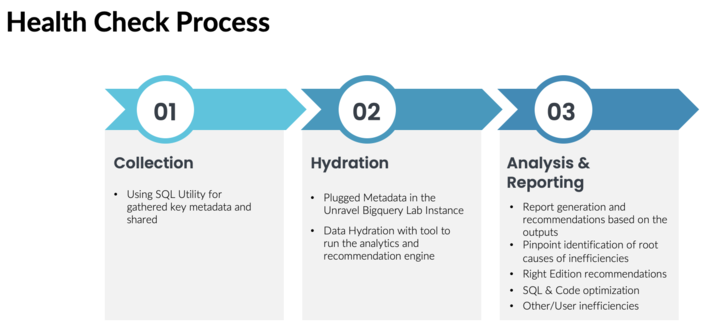
Document Version Record

| Date | Version # | Author | Remarks / Reason |
| --- | --- | --- | --- |
| 05-Jan-23 | 1.0 | Dev Team | New Document |

## Objectives

Health check upload for bigquery unravel product.

## Architecture



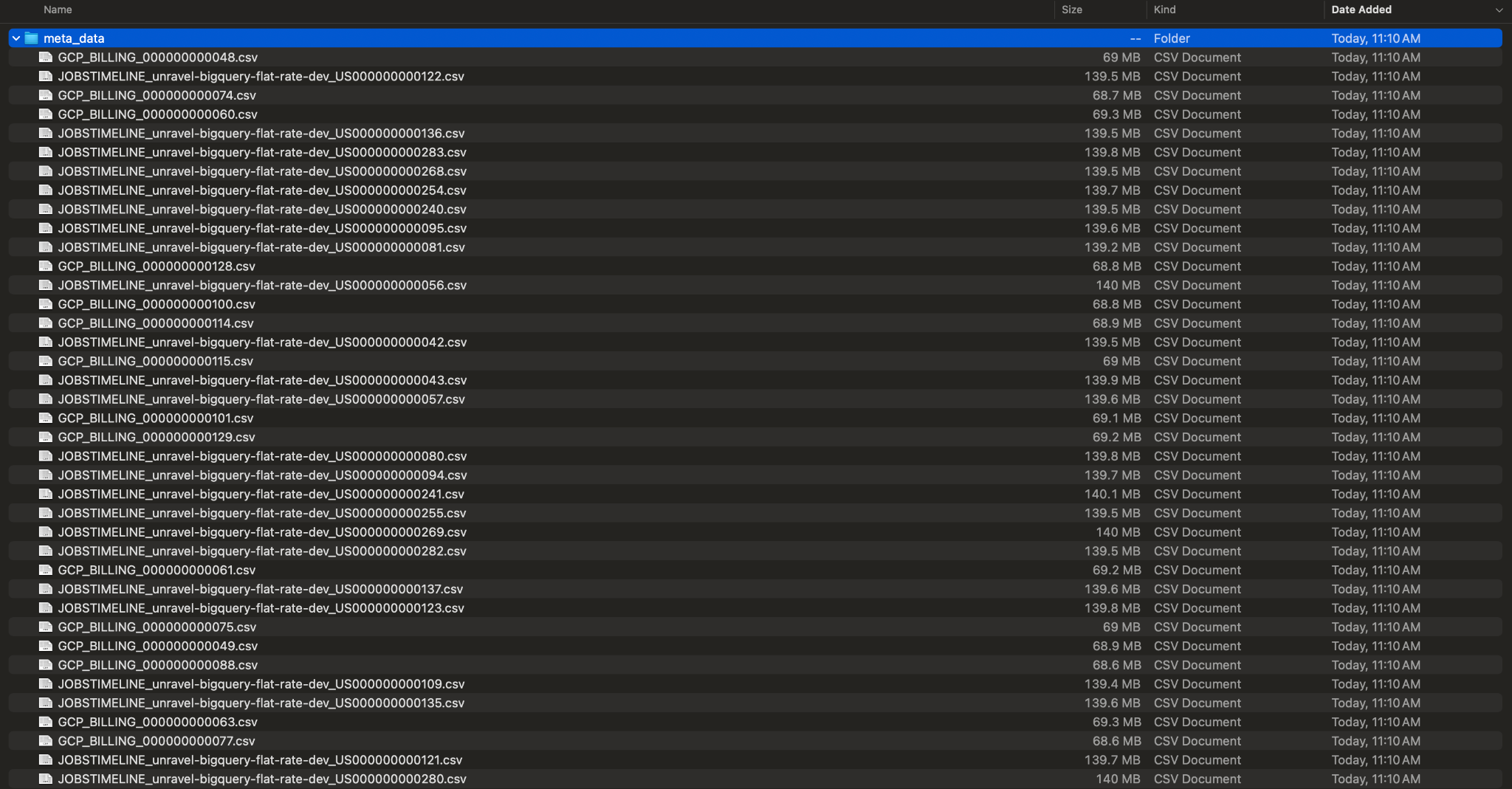
## Prerequisite

1. **BigQuery Permissions/roles required to upload metadata :** 
   1. Permissions required to upload the JOBS, JOBS\_TIMELINE and BILLING metadata to bigquery dataset :
      1. bigquery.datasets.create
      2. bigquery.datasets.get
      3. bigquery.jobs.create
      4. bigquery.tables.create
      5. bigquery.tables.updateData
      6. storage.managedFolders.get
      7. storage.managedFolders.list
      8. storage.objects.get
      9. storage.objects.list

## Upload BigQuery Metadata for health check.

Do the following to upload BigQuery metadata :

1. Upload these csv files to a gcs bucket(s) ([Steps for creating a gcs bucket](#_pv9nzmt9jjd5)) :



1. Clone this repo: <https://github.com/unraveldata-org/BigQuery-data-loader.git>
2. Follow below Steps to create a Service Account credential key that authenticates to upload the metadata to a bigquery dataset :
3. cd terraform/healthcheck
4. cp input.tfvars.example input.tfvars
5. open input.tfvars
6. Under monitoring\_project\_ids add the project id where you want to upload data to bigquery and connect with unravel to perform health check
7. Under svc\_account\_project\_id add the project on which you want the service account to be created to execute the upload script
8. Now, open local.tf
9. Under monitoring\_project\_role\_permission add the below permissions, comma separated (ignore if the same permissions are already present) :
   1. “bigquery.datasets.create”
   2. “bigquery.datasets.get”
   3. “bigquery.jobs.create”
   4. “bigquery.tables.create”
   5. “bigquery.tables.updateData”
   6. “storage.managedFolders.get”
   7. “storage.managedFolders.list”
   8. “storage.objects.get”
   9. “storage.objects.list”
10. Before using this project, you need to authenticate with Google Cloud using gcloud. Follow the instructions provided at <https://cloud.google.com/sdk/docs/install-sdk> for a one-time configuration. You can find the installation instruction based on the Machine Arch and OS installed in the above link.
11. To authenticate gcloud, execute the following commands:
    1. gcloud init
    2. gcloud auth application-default login
12. Now, to run terraform execute the following :
    1. terraform init
    2. terraform plan --var-file=input.tfvars
    3. terraform apply --var-file=input.tfvars
13. Now, there must be a credential file created inside terraform/healthcheck/keys folder, you will need this credential file while providing configuration for the upload script
14. cd Data Uploader Script
15. cp upload\_config.yaml.example upload\_config.yaml
16. cd Data Uploader Script
17. Open upload\_config.yaml file and edit it with your configuration :
18. jobs\_data\_gcs\_bucket\_path: "add the gcs bucket path where you have the csv files of JOBS Data"
19. jobs\_timeline\_data\_gcs\_bucket\_path: "add the gcs bucket path where you have the csv files of JOBSTIMELINE Data"
20. billing\_data\_gcs\_bucket\_path: "add the gcs bucket path where you have the csv files of Billing Data"
21. You can also enter the same gcs bucket path for JOBS, JOBSTIMELINE and BILLING Data
22. upload\_project: "add the project id where you want to upload the data to bigquery for performing Health Check"
23. upload\_dataset: "add the dataset id where you want to upload the data to bigquery for performing Health Check. If the dataset does not exist then a new dataset with this name will be created"
24. jobs\_table\_name: "add the table name which you want to keep for the table where you want to load the JOBS Data"
25. jobs\_timeline\_table\_name: "add the table name which you want to keep for the table where you want to load the JOBS TIMELINE Data"
26. billing\_table\_name: "add the table name which you want to keep for the table where you want to load the BILLING Data"
27. credential: "replace this with path of the credential file created in previous step through terraform"

*Note : Please use a double backslash when defining credential path in download\_config.yaml file for Windows Machine*

*Example :*

*Windows :*

*credential: "C:\\Users\\user1\\path\to\\authentication\\key\\project-1.json"*

*Mac/Linux :*

*credential: "/path/to/authentication/key/project-1.json"*

1. Install required python packages using : pip3 install -r requirements.txt
2. Run upload script with below command :

python3 /path/to/upload\_metadata.py --config\_file /path/to/upload\_config.yaml

1. Tables with JOBS, JOBS\_TIMELINE and BILLING metadata will be created in the defined BigQuery dataset.
2. To destroy the roles and key created by terraform, run :
3. cd ../terraform/healthcheck/
4. terraform destroy --var-file=input.tfvars
5. Now, configure these tables in unravel and start the data hydration.

### Steps for creating a gcs bucket :

1. Go to the Cloud Storage Buckets page: <https://console.cloud.google.com/storage/browser> in the Google Cloud console.
2. Click + Create.
3. On the Create a bucket page, enter the following information:
   1. Name your bucket: Enter a unique name for your bucket. The name must start with a lowercase letter or number, and it can contain up to 63 characters. It can also contain dashes and periods.
   2. Choose where to store your data: Select a location for your bucket. You can choose a location in the same region as your project, or you can choose a different region.
   3. Choose a storage class for your data: Select a storage class for your bucket. The storage class determines how long your data is kept and how much it costs to store.
4. Click Create.
5. Create new folder in bucket

## Video(s) :

* <https://drive.google.com/file/d/1-J2z6hGpSwGB0Dbv1o8jd5V4tDuilIWU/view?usp=sharing>
* <https://drive.google.com/file/d/13trMfsay4S-lanw5KqvqotDwhdsleDCs/view?usp=sharing>