### BigQuery Audit Logs Overview

<https://cloud.google.com/bigquery/docs/reference/auditlogs>

[BigQueryAuditMetadata](https://cloud.google.com/bigquery/docs/reference/auditlogs#bigqueryauditmetadata_examples): The new version of logs, which reports resource interactions such as which tables were read from and written to by a given query job and which tables expired due to having an expiration time configured.  
**In general, users will want to leverage the new BigQueryAuditMetadata logs functionality.**

Audit logs are organised into the following three streams. For more information about the streams, see the [Cloud Audit Logs](https://cloud.google.com/logging/docs/audit) documentation.

* **Data access** - The data\_access stream contains entries about jobs by using the JobInsertion and JobChange events and about table data modifications by using the TableDataChange and TableDataRead events.
* **System event** - You can set an expiration time on tables to remove them at a specified time. The system\_event stream reports a TableDeletion event when the table expires and is removed.
* **Admin activity -** Creating, deleting, and updating resources related to [BigQuery Reservations](https://cloud.google.com/bigquery/docs/reservations-concepts) are reported in the admin activity stream.

### Methods

* protoPayload.methodName is set to one of the following values, highlighted ones are captured for POC use case:
  + google.cloud.bigquery.v2.TableService.InsertTable
  + google.cloud.bigquery.v2.TableService.UpdateTable
  + google.cloud.bigquery.v2.TableService.PatchTable
  + google.cloud.bigquery.v2.TableService.DeleteTable
  + google.cloud.bigquery.v2.DatasetService.InsertDataset
  + google.cloud.bigquery.v2.DatasetService.UpdateDataset
  + google.cloud.bigquery.v2.DatasetService.PatchDataset
  + google.cloud.bigquery.v2.DatasetService.DeleteDataset
  + google.cloud.bigquery.v2.TableDataService.List
  + google.cloud.bigquery.v2.JobService.InsertJob
  + google.cloud.bigquery.v2.JobService.Query
  + google.cloud.bigquery.v2.JobService.GetQueryResults
  + InternalTableExpired

### Events captured in Metadata

* Highlighted ones are captured for POC use case:
  + jobInsertion
  + jobChange
  + jobDeletion
  + datasetCreation
  + datasetChange
  + datasetDeletion
  + tableCreation
  + tableChange
  + tableDeletion
  + tableDataRead
  + tableDataChange
  + modelDeletion
  + modelCreation
  + modelMetadataChange
  + modelDataChange
  + modelDataRead
  + routineCreation
  + routineChange
  + routineDeletion
  + rowAccessPolicyCreation
  + rowAccessPolicyChange
  + rowAccessPolicyDeletion

### Create a BQ AuditLog Sink

gcloud logging sinks create *<my-example-sink>* bigquery.googleapis.com/projects/<*my-project-id>*/datasets/<*auditlog\_dataset>* \

--log-filter='protoPayload.metadata."@type"="type.googleapis.com/google.cloud.audit.BigQueryAuditMetadata"'

References:

<https://cloud.google.com/bigquery/docs/reference/auditlogs#defining_a_bigquery_log_sink_using_gcloud>

<https://cloud.google.com/sdk/gcloud/reference/logging/sinks/create>

### Activity Log Example

Below is a sample of how an activity log entry looks like, **highlighted** text is the details we are capturing for BQ Audit log report.

{

"logName": "projects/gcp-project/logs/cloudaudit.googleapis.com%2Factivity",

"resource": {

"type": "bigquery\_dataset",

"labels": {

**"dataset\_id": "temp\_tables",**

**"project\_id": "gcp-project"**

}

},

"protopayload\_auditlog": {

"serviceName": "bigquery.googleapis.com",

**"methodName": "google.cloud.bigquery.v2.TableService.UpdateTable",**

**"resourceName": "projects/gcp-project/datasets/temp\_tables/tables/test\_bq\_view",**

"resourceLocation": null,

"numResponseItems": null,

"status": null,

"authenticationInfo": {

**"principalEmail": "ygupta@xyz.com",**

"authoritySelector": null,

"serviceAccountKeyName": null,

"serviceAccountDelegationInfo": [],

"principalSubject": null

},

"authorizationInfo": [{

"resource": "projects/gcp-project/datasets/temp\_tables/tables/test\_bq\_view",

"permission": "bigquery.tables.update",

"granted": "true",

"resourceAttributes": null

}],

"policyViolationInfo": null,

"requestMetadata": {

"callerIp": "120.17.107.255",

"callerSuppliedUserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36,gzip(gfe)",

"callerNetwork": null,

"requestAttributes": null,

"destinationAttributes": null

},

**"metadataJson":** "{\"@type\":\"type.googleapis.com/google.cloud.audit.BigQueryAuditMetadata\",\"**tableChange**\":{\"reason\":\"TABLE\_UPDATE\_REQUEST\",\"table\":{\"createTime\":\"2022-07-22T04:48:24.052Z\",\"expireTime\":\"2022-08-21T04:48:24.052Z\",\"schemaJson\":\"{\\n \\\"fields\\\": [{\\n \\\"name\\\": \\\"crn\\\",\\n \\\"type\\\": \\\"STRING\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }, {\\n \\\"name\\\": \\\"lylty\_card\_nbr\\\",\\n \\\"type\\\": \\\"STRING\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }, {\\n \\\"name\\\": \\\"lylty\_cust\_type\_desc\\\",\\n \\\"type\\\": \\\"STRING\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }, {\\n \\\"name\\\": \\\"lylty\_card\_status\_desc\\\",\\n \\\"type\\\": \\\"STRING\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }, {\\n \\\"name\\\": \\\"last\_scan\_date\\\",\\n \\\"type\\\": \\\"DATE\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }, {\\n \\\"name\\\": \\\"reward\_points\\\",\\n \\\"type\\\": \\\"NUMERIC\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }, {\\n \\\"name\\\": \\\"reward\_balance\\\",\\n \\\"type\\\": \\\"NUMERIC\\\",\\n \\\"mode\\\": \\\"NULLABLE\\\"\\n }]\\n}\",\"tableInfo\":{\"description\":\"test\"},\"tableName\":\"projects/gcp-project/datasets/temp\_tables/tables/test\_bq\_view\",\"updateTime\":\"2022-07-22T04:49:06.637Z\",\"view\":{\"**query**\":\"**SELECT \* FROM `gcp-project.adhoc\_analysis.aging\_card\_testing` LIMIT 10**\"}}}}"

},

"textPayload": null,

"timestamp": "2022-07-22T04:49:06.682Z",

**"receiveTimestamp": "2022-07-22T04:49:07.616Z",**

**"severity": "NOTICE",**

"insertId": "9ppsv6dj351",

"httpRequest": null,

"operation": null,

"trace": null,

"spanId": null,

"traceSampled": null,

"sourceLocation": null,

"split": null

}

### SQL Query - Log Extraction

WITH audit\_log as (

SELECT

resource.labels.dataset\_id,

resource.labels.project\_id,

--protopayload\_auditlog.methodName,

REGEXP\_EXTRACT('google.cloud.bigquery.v2.TableService.UpdateTable',r'.\*\.([^/$]\*)(?:\.)') as service,

REGEXP\_EXTRACT(protopayload\_auditlog.methodName,r'.\*\.([^/$]\*)') as method,

--protopayload\_auditlog.resourceName,

**REGEXP\_EXTRACT(protopayload\_auditlog.resourceName,r'.\*tables\/([^/$]\*)') as tableName,**

REGEXP\_CONTAINS(protopayload\_auditlog.authenticationInfo.principalEmail,r'.+(@woolworths\.com\.au)') as is\_ww\_email,

protopayload\_auditlog.authenticationInfo.principalEmail,

protopayload\_auditlog.metadataJson,

COALESCE(JSON\_EXTRACT(JSON\_EXTRACT(JSON\_EXTRACT(JSON\_EXTRACT(protopayload\_auditlog.metadataJson, "$.tableCreation"),"$.table"),"$.view"),"$.query"), JSON\_EXTRACT(JSON\_EXTRACT(JSON\_EXTRACT(JSON\_EXTRACT(protopayload\_auditlog.metadataJson, "$.tableChange"),"$.table"),"$.view"),"$.query"), JSON\_EXTRACT(JSON\_EXTRACT(protopayload\_auditlog.metadataJson, "$.tableDeletion"),"$.reason") ) as query,

case when JSON\_EXTRACT(protopayload\_auditlog.metadataJson, "$.tableCreation") is not null then 'tableCreation'

when JSON\_EXTRACT(protopayload\_auditlog.metadataJson, "$.tableChange") is not null then 'tableChange'

when JSON\_EXTRACT(protopayload\_auditlog.metadataJson, "$.tableDeletion") is not null then 'tableDeletion'

else 'N/A' end as type,

receiveTimestamp

**FROM `gcp-project.bq\_auditlog.cloudaudit\_googleapis\_com\_activity\_\*` -- Get data from all table partitions**

**WHERE DATE(timestamp) >= "2022-07-10"**

**and severity = 'NOTICE'-- Exclude ERROR**

**--and protopayload\_auditlog.authenticationInfo.principalEmail like '%ygupta%'**

**and protopayload\_auditlog.methodName in**

**('google.cloud.bigquery.v2.TableService.PatchTable',**

**'google.cloud.bigquery.v2.TableService.UpdateTable',**

**'google.cloud.bigquery.v2.TableService.InsertTable',**

**'google.cloud.bigquery.v2.JobService.InsertJob',**

**'google.cloud.bigquery.v2.TableService.DeleteTable',**

**'google.cloud.bigquery.v2.JobService.Query' )**

)

select

dataset\_id,

project\_id,

service,

method || '-' || type as change\_type,

tableName,

is\_ww\_email,

principalEmail,

query,

receiveTimestamp,

ROW\_NUMBER() over (partition by dataset\_id,project\_id,tableName order by receiveTimestamp desc) as change\_order,

ROW\_NUMBER() over (partition by dataset\_id,project\_id,tableName,type,principalEmail order by receiveTimestamp asc) as change\_order\_table\_creation,

ROW\_NUMBER() over (partition by dataset\_id,project\_id,tableName,type,principalEmail order by receiveTimestamp desc) as change\_order\_table\_creation\_reverse,

metadataJson,

case when query = LAG(query) OVER (PARTITION BY project\_id,dataset\_id,tableName, case when type = 'tableDeletion' then 1 else 0 end ORDER BY receiveTimestamp ASC) then 'No'

when LAG(query) OVER (PARTITION BY project\_id,dataset\_id,tableName,case when type = 'tableDeletion' then 1 else 0 end ORDER BY receiveTimestamp ASC) is NULL then 'N/A'

else 'Yes' end as query\_changed,

LAST\_VALUE(is\_ww\_email) OVER (PARTITION BY project\_id,dataset\_id,tableName,type ORDER BY receiveTimestamp ASC RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING) AS last\_created\_by\_mail\_type

FROM audit\_log

where type != 'N/A'

order by receiveTimestamp asc

### SQL Query - Flag Calculations

Below flags will be used in conjunction with DataStudio report page filters to get the relevant data.

*ROW\_NUMBER() over (partition by dataset\_id,project\_id,tableName order by receiveTimestamp desc) as* ***change\_order****,*

– Tag the latest log entry record for every table as 1 ( so we can filter it out later to get last updated by)

*ROW\_NUMBER() over (partition by dataset\_id,project\_id,tableName order by case when type = 'tableCreation' then 1 else 0 end desc, receiveTimestamp asc) as* ***change\_order\_table\_creation****,*

– Tag the first log entry record for every table as 1, with priority to ‘tableCreation event’. In case the first log entry is something other than tableCreation, it will be ignored later in Datastudio report. This flag will help us get the first instance when this table was created

*case when query = LAG(query) OVER (PARTITION BY project\_id,dataset\_id,tableName ORDER BY receiveTimestamp ASC) then 'No'*

*when LAG(query) OVER (PARTITION BY project\_id,dataset\_id,tableName ORDER BY receiveTimestamp ASC) is NULL then 'N/A'*

*else 'Yes' end as* ***query\_changed****,*

– Applicable only to views ( Datastudio report will filter the whole page for view only). Check if the query in current log entry is same as last one. In case of deletions LAG(Query) will be NULL which will result in N/A.

*LAST\_VALUE(is\_ww\_email) OVER (PARTITION BY project\_id,dataset\_id,tableName,type ORDER BY receiveTimestamp ASC RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING) AS* ***last\_created\_by\_mail\_type***

– For every tableCreation, get the latest log entry and extract the email that was used for latest tableCreation.If the email is same as the email that first created the table (by same I mean a WW Email) then its being created manually, if not - that would mean the ownership shifted to Service Account

### 