Introduction to BigLake tables

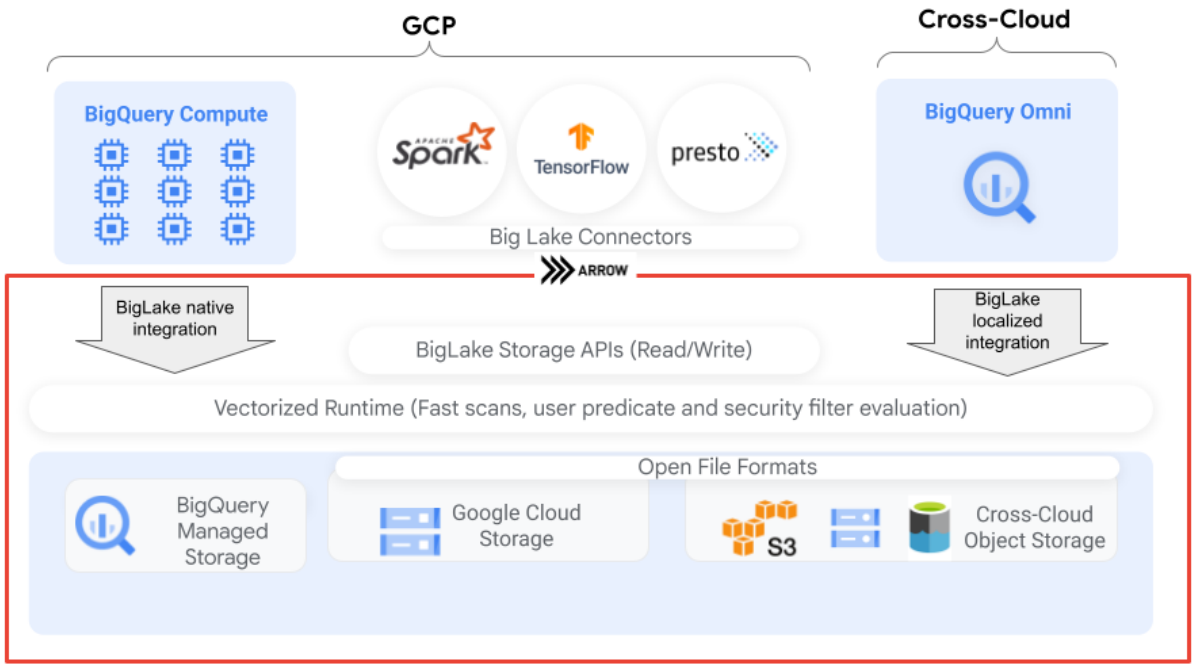
This document provides an overview of BigLake and assumes familiarity with database tables and permissions management. For directions on using BigLake tables, see Create and manage BigLake tables (/bigquery/docs/biglake-quickstart).

Overview

BigLake is a unified storage engine that simplifies data access for data warehouses and lakes by providing uniform fine-grained access control across multi-cloud storage and open formats.

BigLake extends BigQuery's fine-grained row- and column- level security (including dynamic data masking (/bigquery/docs/column-data-masking)) to tables on data resident object stores such as Amazon S3, Azure Data Lake Storage Gen2, and Cloud Storage. BigLake decouples access to the table from the underlying cloud storage data through access delegation. This feature helps you to securely grant row- and column-level access to users and pipelines in your organization without providing them full access to the table.

After you create a BigLake table, you can query it like other BigQuery tables. BigQuery enforces row- and column-level access controls. Every user sees only the slice of data that they are authorized to see. Governance policies are enforced on all access to the data through BigQuery APIs. For example, the BigQuery Storage API (/bigquery/docs/reference/storage) lets users access authorized data using open source query engines such as Apache Spark, as the following diagram shows:



BigLake tables on object stores

For data administrators, BigLake lets you abstract access management on data lakes from files to tables, and it helps you manage users' access to data on lakes.

Because BigLake tables on object stores are designed to simplify the access model for tables that are connected to object stores, we recommend using BigLake tables to build and maintain connections to these object stores.

You can use external tables (/bigquery/docs/external-tables) in cases where governance is not a requirement, or for ad hoc data discovery and manipulation.

Limitations

- BigLake tables on object stores are subject to the same limitations as BigQuery tables. For more information, see Quotas (/bigquery/quotas#external_tables).
- BigLake does not support downscoped credentials from Dataproc Personal Cluster Authentication (/dataproc/docs/concepts/iam/personal-auth). As a workaround, to use clusters with Personal Cluster Authentication, you must inject your credentials using an empty Credential Access Boundary (https://cloud.google.com/dataproc/docs/concepts/iam/personal-auth#create_a_cluster_and_enable_an_interactive_session)

with the `--access-boundary=<(echo -n "{}")` flag. For example, the following command enables a credential propagation session in a project named `myproject` for the cluster named `mycluster`:

```
$ gcloud dataproc clusters enable-personal-auth-session \
    --region=us \
    --project=myproject \
    --access-boundary=<(echo -n "{}") \
    mycluster
```

Caution: Using an empty credential access boundary removes one layer of protection against attacks through stolen credentials from Dataproc clusters. Stolen credentials have a larger blast radius without downscoping.

As an alternative, you can disable Personal Cluster Authentication and use the Dataproc virtual machine (VM) service account (/dataproc/docs/concepts/configuring-clusters/service-accounts) as a proxy for user groups.

- BigLake tables are read-only. You cannot modify BigLake tables using DML statements or other methods.
- BigLake tables support the following five formats:
  - Avro
  - CSV
  - JSON
  - ORC
  - Parquet
- The BigQuery Storage API (/bigquery/docs/reference/storage) is not available in other cloud environments, such as AWS and Azure.

Security model

This guidance is intended for the following organizational roles:

- **Data lake administrators.** These administrators typically manage Identity and Access Management (IAM) policies on Cloud Storage buckets and objects.

- **Data warehouse administrators.** These administrators typically create, delete, and update BigLake tables. Data warehouse administrators need the following IAM roles (/iam/docs/understanding-roles#predefined_roles):
  - BigQuery Admin or BigQuery Data Owner
  - BigQuery Connection Admin
- **Data analysts.** Analysts typically have the BigQuery User role and can read data and run queries.

Data lake administrators are responsible for granting read privileges to connections that data warehouse administrators manage. In turn, data warehouse administrators define BigLake tables, set appropriate access controls (such as column and row security), and share the BigLake tables with data analysts.

Caution: Data analysts should **not** have the following:

- The ability to read objects directly from Cloud Storage (see the Storage Object Viewer IAM role (/storage/docs/access-control/iam-roles)), which lets data analysts circumvent access controls placed by data warehouse administrators.
- The ability to bind tables to connections (like the BigQuery Connection Administrator).

  Otherwise, data analysts can create new BigLake tables that do not have any access controls, thus circumventing controls placed by data warehouse administrators.

BigLake tables with Analytics Hub

BigLake tables are compatible with Analytics Hub. Datasets containing BigLake tables can be published as Analytics Hub listings (/bigquery/docs/analytics-hub-introduction#listings). Analytics Hub subscribers can subscribe to these listings, which provision a read-only dataset, called a *linked dataset* (/bigquery/docs/analytics-hub-introduction#linked_datasets), in their project. Subscribers can query all tables in the linked dataset, including all BigLake tables. For more information, see Subscribe to a listing (/bigquery/docs/analytics-hub-view-subscribe-listings#subscribe_to_a_listing).

BigQuery ML with BigLake tables

You can use BigQuery ML (/bigquery-ml/docs/introduction) to train and run models on BigLake in Cloud Storage.

What's next

- Learn how to create and manage BigLake tables (/bigquery/docs/biglake-quickstart).

Last updated 2022-10-03 UTC.