BigQuery and Google Cloud IAM

Welcome to the "BigQuery and Google Cloud IAM" module. This module summarizes the key details of the Google Cloud Identity and Access Management (IAM) model, including how roles and permissions are applied to datasets and tables in BigQuery. Drawing upon your knowledge of Redshift, this module also provides a high-level overview of the similarities and differences in roles and permissions between Redshift and BigQuery to help you start securing and sharing your data in BigQuery.

Let's jump in!

Lesson 1 of 19

BigQuery and Google Cloud IAM

Upon completion of this module, you will be able to:

1. Describe Google Cloud IAM roles and permissions for BigQuery.
2. Explain how the roles and permissions in Redshift differ from BigQuery.
3. Implement Google Cloud IAM roles and permissions for BigQuery.
4. Secure and share datasets and tables in BigQuery.

Lesson introduction

In this lesson, you will examine the key similarities and differences between Redshift and BigQuery regarding the roles and permissions that are used to control access to data.

Let's get started!

Key points about roles and permissions

Let's start with some ways that roles and permissions in Redshift and BigQuery are similar.

- Redshift and BigQuery both have fine-grained access controls that let you control access down to the row and column level. Permissions are also inherited using the resource hierarchy.
- Redshift and BigQuery both have encryption for data at rest and provide automated tokenization of data when needed.

BigQuery's AEAD encryption provides a method to combine data from different entities in the same table without revealing the data to other parties.

The following differences are helpful to understand.

Click the + icon to expand and learn more.

Third-party authentication

- In addition to using AWS Identity and Access Management (IAM) roles, Redshift allows third-party identity providers to authenticate directly to a Redshift database.
- BigQuery uses Google Cloud IAM to authenticate third party providers.

Permissions for compute resourcing

- Redshift Serverless has a permissions role for serverless compute resource management. You need this role to make changes to the default compute allocation of Redshift Processing Units (RPUs).
- To run jobs in BigQuery, you need the permission 'bigquery.jobs.create.' If you are using on-demand billing, you do not have to manually allocate slot resources; if you are using a different billing plan such as flat-rate, then you need additional permissions to be able to allocate slots.

Object ownership

- In Redshift, the user account that creates a database object is automatically made the owner with full ownership permissions.
- Permissions in BigQuery are managed using IAM and inheritance from higher levels in the resource hierarchy, including the BigQuery Data Owner role.

Lesson 4 of 19

BigQuery and Redshift

Here's a summary of access control, encryption, and network security options in BigQuery and Redshift.

Topic	BigQuery	Redshift
Access control hierarchy	Based on Google Cloud Identity and Access Management with access control options at all levels of the resource hierarchy: • Organization level • Folder level (optional) • Project level • Dataset level • Table level • Column level • Row level Summary of BigQuery data governance and security	High-level (cluster or above) access control using AWS IAM, low-level (DB or lower) granular access control through RBAC (uses SQL users, roles, grants, privileges assigned via DDL) Redshift security overview
Access control at highest organizational level	Organizations manage Projects, which contain BigQuery datasets.	Namespaces for Redshift Serverless

Topic	BigQuery	Redshift
Access control at table level	Table level	Based on the database level
Access control at column level	Column-level access control	Column-level security
Access control at row level	Row-level security	Row-level security
Encryption	Encryption at rest by default Customer managed encryption keys AEAD encryption Column-level encryption	Encryption at rest Column data tokenization (similar to column-level encryption in BigQuery)
Network security	Private Google Access or Private Services Connect Virtual Private Cloud (VPC) Service Controls VPC Network Peering	Sharing data across clusters Private connectivity using VPC endpoint Security groups for Firewall

Next, let's examine these topics in BigQuery with more detail.

Lesson 5 of 19

Lesson introduction

In this lesson, you will learn how you can use Google Cloud's Identity and Access Management (IAM) to start securing your resources and data and learn about other Google Cloud security features for authentication, encryption, and network connectivity.

Let's get started!

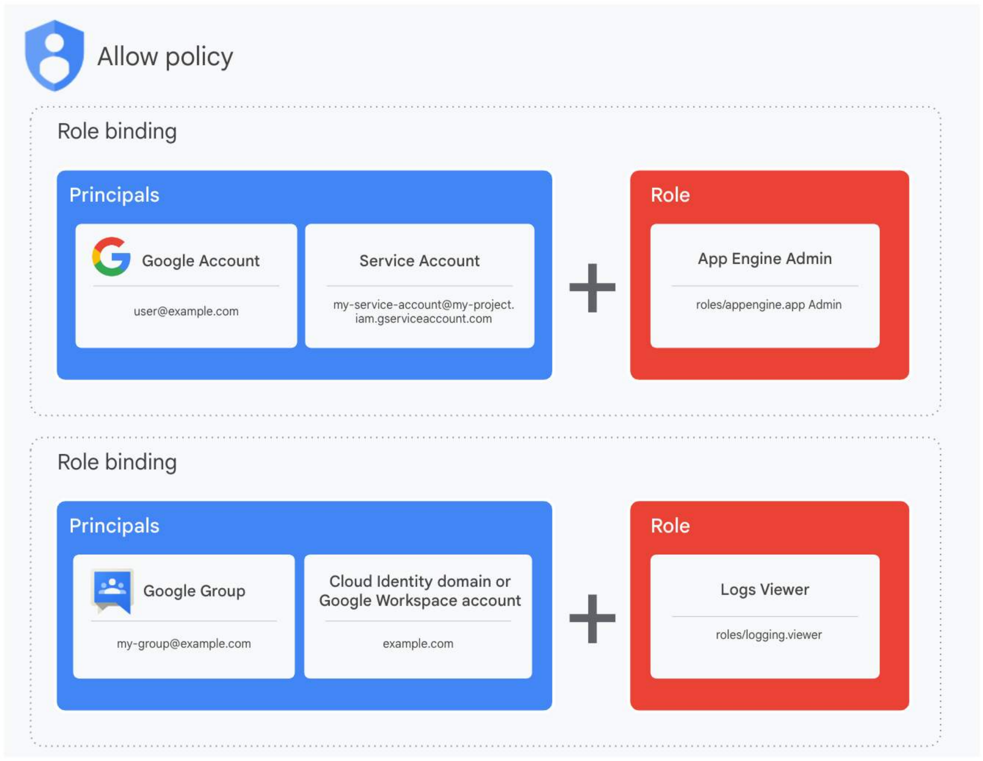
What is IAM?

In Google Cloud, IAM is a tool and framework for fine-grained access control and visibility for centrally managing cloud resources. It provides a simple and consistent access control interface for all Google Cloud services and is provided at no cost to all Google Cloud customers.

IAM enables you to grant granular access to specific Google Cloud resources and to prevent unauthorized access to other resources as needed.

IAM empowers you adopt the **security principle of least privilege**, which states that users or tools should have only the specific access and permissions that they actually need to complete a given task.

IAM follows a model of access management composed of three main parts: Principal, Role, and Policy.



When you want to define who (principal) has what type of access (role) on a resource, you create an *allow* policy and attach it to the resource such as a BigQuery dataset.

Click the + icon to expand and learn more.

Principal —

A principal can be a Google Account (for end users), a service account (for applications and compute workloads), a Google group, a Google Workspace account, or Cloud Identity domain that can access a resource.

Each principal has its own identifier, which is typically an email address.

Role —

A role is a collection of permissions. Permissions determine which operations are allowed on a resource.

When you grant a role to a principal, you grant all the permissions that the role contains.

Policy —

The *allow* policy is a collection of role bindings that bind one or more principals to individual roles.

In IAM, permission to access a resource is not granted directly to the end user. Instead, permissions are grouped into roles, and roles are granted to authenticated principals.

An *allow* policy, also known as an IAM policy, defines and enforces what roles are granted to which principals. Each *allow* policy is attached to a Google Cloud resource such as a BigQuery dataset. When an authenticated principal attempts to access a resource, IAM checks the resource's *allow* policy to determine whether the action is permitted.

Lesson 7 of 19

Additional security features

How does data encryption work on Google Cloud?

By default, all data stored within Google Cloud is encrypted at rest and when in transit to and from outside of Google Cloud, using the same key management systems that Google uses for its own encrypted data. These key management systems provide strict key access controls and auditing, and encrypt user data at rest using AES-256 encryption standards. No setup, configuration, or management is required.

There are also many user-configured and customer-managed encryption options for data at rest, in transit, and in use.

Click the + icon to expand and learn more.

Encryption at rest

Default encryption at rest protects your data from a system compromise or data exfiltration by encrypting data while stored.

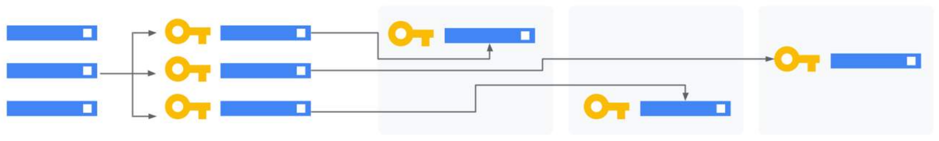
For example, BigQuery data is divided into several chunks, each with its own data encryption key. Keys are automatically stored and managed in Cloud Key Management Service (KMS). You can also manage your own keys using customer-managed encryption keys (CMEK) in Cloud KMS.

Encryption in transit

Default encryption in transit to and from outside of Google Cloud protects your data if communications are intercepted while data moves between your site and Google Cloud. This protection is achieved by encrypting the data before transmission, authenticating the endpoints, and, on arrival, decrypting and verifying that the data was not modified.

Encryption in use

Google Cloud offers Confidential Computing, which you can use to secure your most sensitive data by keeping it encrypted in use, while it is being processed.



When data is uploaded to BigQuery, it is broken up into chunks, each chunk with its own encryption key. Both the encrypted chunks and wrapped encryption keys are then distributed across Google's secure storage infrastructure.

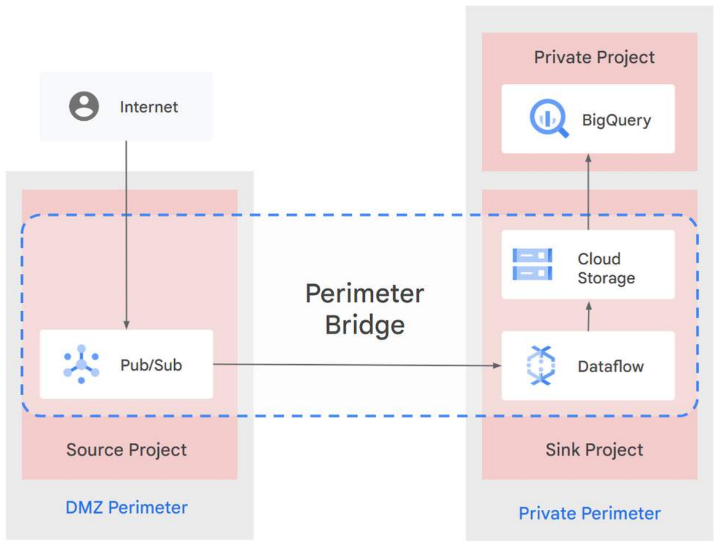
How does network security work for connections to Google Cloud?

Many organizations have workloads both in cloud environments and in on-premises environments. In addition, for resiliency, some organizations use multicloud solutions. In these scenarios, it's critical to secure your connectivity between all of your environments.

Google Cloud includes private access methods for virtual machines (VMs) that are supported by Cloud Virtual Private network (VPN) or Cloud Interconnect, including the following:

- Use IPSec VPNs to link your Virtual Private Cloud (VPC) networks to your on-premises data center or to other cloud providers.

- Use Dedicated Interconnect and Partner Interconnect to link your VPC networks to your on-premises data center or to other cloud providers over high-speed direct connections.
- Use Private Service Connect for services to access service endpoints that are provided by your organization or by another provider.
- Use Private Service Connect to let VMs access Google application programming interface (APIs) using internal IP addresses. With Private Service Connect, your VMs don't have to have external IP addresses in order to access Google services.



A perimeter bridge allows projects in different service perimeters to communicate. For example, a source project "A" in the DMZ perimeter and a sink project "B" in the private perimeter can share a perimeter bridge that allows data to flow between the two projects. The sink project "B" can also share perimeter bridge with a private project "C" in the private perimeter. In this case, data can move between the source and sink projects ("A" and "B") and the sink and private projects ("B" and "C"), but not directly between the source and private project ("A" and "C").

References

- Learn more about Google Cloud IAM roles and permissions from the documentation page titled "IAM policy inheritance."
- Learn more about how IAM policies are applied and inherited across the resource hierarchy from the documentation titled "Resource hierarchy."
- Learn about Google's BeyondCorp initiative and security framework based on Zero Trust security from the documentation titled "Security framework."
- Learn more about encryption on Google Cloud from the documentation titled "Encryption."
- Learn more about authentication on Google Cloud from the documentation titled "Authentication at Google."
- Learn more about Google Cloud best practices for securing your network from the documentation titled "Secure your network."

Lesson 9 of 19

Lesson introduction

In this lesson, you will learn more about Google Cloud IAM roles and permissions for BigQuery, including how identity and access management policies are inherited by BigQuery resources and how to leverage predefined roles for BigQuery.

Let's get started!

Lesson 10 of 19

IAM policies and permission inheritance

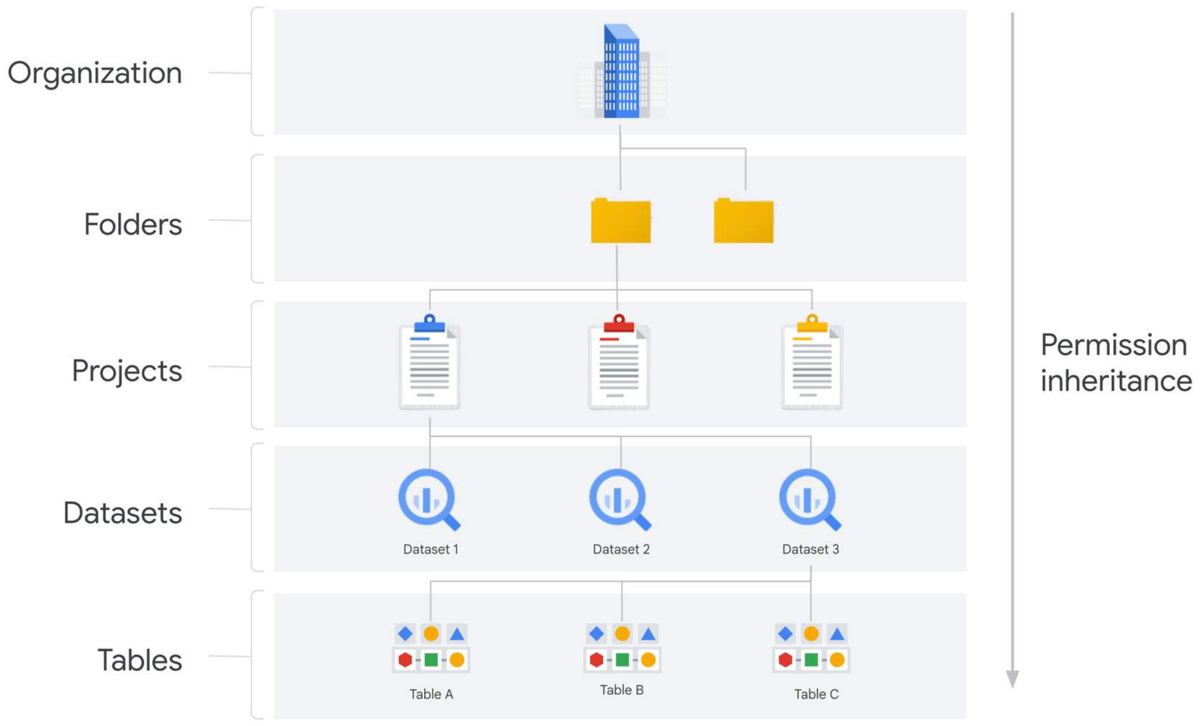
How does the IAM and BigQuery permission hierarchy impact your users?

The Google Cloud IAM model of principals, roles, and policies is designed to fit the needs of most organizations and projects. It is highly likely that there is a predefined role for the majority of your requirements. However, in cases where there is not, it is possible to create a custom role and to assign that role the specific permissions needed.

Imagine you want to give your entire team access to specific BigQuery data. You might store that data in one or more datasets on a single Google Cloud project to simplify access management.

BigQuery tables inherit the permissions of their parent dataset. BigQuery datasets inherit permissions from their parent entities in the resource hierarchy (starting with organizations down to folders and projects).

To perform an operation on a BigQuery resource, a user needs both the relevant permissions on the resource, and also permission to execute a BigQuery job. The permission to run a job needs to be assigned within the project that is used to run that job.

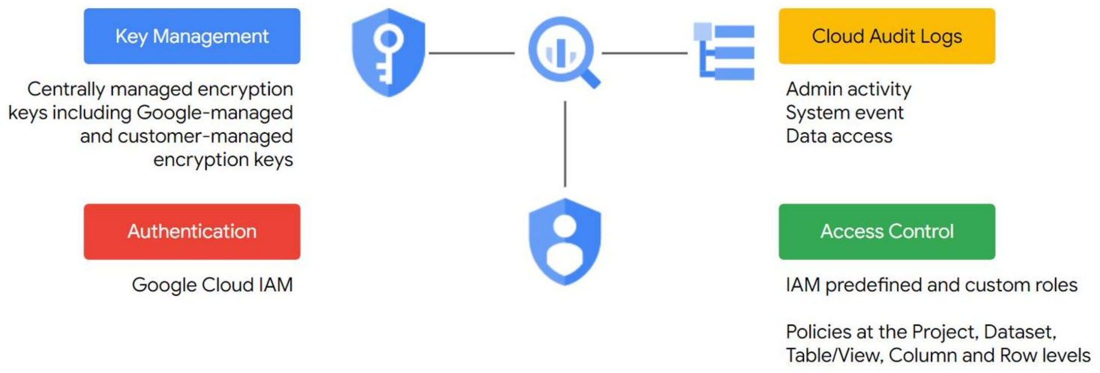


In Google Cloud, granting an IAM permission to a parent resource impacts all of the child resources below it, including BigQuery datasets and tables.

Lesson 11 of 19

Google Cloud and BigQuery integration

How does Google Cloud support security, encryption, and auditing in BigQuery?


Key Management
Centrally managed encryption keys including Google-managed and customer-managed encryption keys
Authentication
Google Cloud IAM
Cloud Audit Logs
Admin activity
System event
Data access
Access Control
IAM predefined and custom roles
Policies at the Project, Dataset, Table/View, Column and Row levels

In BigQuery, access control derives from Google Cloud IAM roles. Permissions are inherited from the organization, folder, and project levels, in addition to any specific BigQuery roles that are applied.

In BigQuery, third-party identity providers cannot authenticate directly to the database. To support authentication from a third-party provider, you must configure the provider as an identity in Google Cloud, and create and assign an IAM role for it with the appropriate permissions.

Access control roles can be applied in BigQuery at the level of datasets, tables, views, or columns. Row level access can also be secured through row-level permissions in BigQuery.

Auditability is based on immutable BigQuery logs available within Google's Cloud Operations suite of tools such as Cloud Monitoring and Cloud Logging. BigQuery admin activities and system events such as table expirations are also logged.

BigQuery storage encrypts data at rest and in transfer using Google-managed encryption keys. You can also leverage customer-managed encryption keys with Cloud KMS.

BigQuery audit logs

Audit logs help you answer "who did what, where, and when?" within Google Cloud resources, including BigQuery.

BigQuery provides audit logs organized into three streams:

- Data access: entries about query jobs and table data modifications
- System event: entries on events such as table expirations
- Admin activity: entries on reservations for commitments and all remaining activities and events such as table and dataset creation

Export BigQuery audit logs from Cloud Logging to a BigQuery dataset

Exporting audit logs from Cloud Logging to another secure storage location, including BigQuery or Cloud Storage, enables you to have a longer-term retention period. By exporting the logs directly to BigQuery, you can run SQL queries on the logs, enabling complex filtering for analysis. Additionally, you can create dashboards using visualization tools, such as Looker Studio.

Assigning roles in BigQuery

Using predefined roles

An important aspect of operating a data warehouse is allowing shared but controlled access against the same data to different groups of users.

Many traditional data warehouses accomplish this using row-level security. You can achieve the same results in BigQuery with access control to datasets, tables, views, or columns, or by defining authorized views and row-level permissions.

IAM provides predefined roles for BigQuery that can be applied to datasets, tables, views, or columns for fine-grained access control. You can also create custom Cloud IAM roles consisting of your defined set of permissions, and then assign those roles to users or groups.

Here are a few key predefined IAM roles for BigQuery.

Click the + icon to expand and learn more.

BigQuery Admin

(roles/bigquery.admin)

Provides permissions to manage all resources within the project. Can manage all data within the project, and can cancel jobs from other users running within the project.

Lowest-level resources where you can grant this role:

- Datasets
- Row access policies
- Tables
- Views

BigQuery Data Editor

(roles/bigquery.dataEditor)

When applied to a table or view, this role provides permissions to:

- Read and update data and metadata for the table or view.
- Delete the table or view.

This role cannot be applied to individual models or routines.

When applied to a dataset, this role provides permissions to:

- Read the dataset's metadata and list tables in the dataset.
- Create, update, get, and delete the dataset's tables.

When applied at the project or organization level, this role can also create new datasets.

Lowest-level resources where you can grant this role:

- Table
- View

BigQuery Data Owner

(roles/bigquery.dataOwner)

When applied to a table or view, this role provides permissions to:

- Read and update data and metadata for the table or view.
- Share the table or view.
- Delete the table or view.

This role cannot be applied to individual models or routines.

When applied to a dataset, this role provides permissions to:

- Read, update, and delete the dataset.
- Create, update, get, and delete the dataset's tables.

When applied at the project or organization level, this role can also create new datasets.
Lowest-level resources where you can grant this role:

- Table
- View

BigQuery Data Viewer

(roles/bigquery.dataViewer)

When applied to a table or view, this role provides permissions to:

- Read data and metadata from the table or view.

This role cannot be applied to individual models or routines.
When applied to a dataset, this role provides permissions to:

- Read the dataset's metadata and list tables in the dataset.
- Read data and metadata from the dataset's tables.

When applied at the project or organization level, this role can also enumerate all datasets in the project. Additional roles, however, are necessary to allow the running of jobs.

Lowest-level resources where you can grant this role:

- Table
- View

BigQuery Filtered Data Viewer

(roles/bigquery.filteredDataViewer)

Access to view filtered table data defined by a row access policy.

BigQuery Job User

(roles/bigquery.jobUser)

Provides permissions to run jobs, including queries, within the project.

Lowest-level resources where you can grant this role:

- Project

BigQuery Resource Admin

(roles/bigquery.resourceAdmin)

Administer all BigQuery resources.

BigQuery User

(roles/bigquery.user)

When applied to a dataset, this role provides the ability to read the dataset's metadata and list tables in the dataset.

When applied to a project, this role also provides the ability to run jobs, including queries, within the project. A principal with this role can enumerate their own jobs, cancel their own jobs, and enumerate datasets within a project. Additionally, this role allows the creation of new datasets within the project; the creator is granted the BigQuery Data Owner role **(roles/bigquery.dataOwner)** on these new datasets.

Lowest-level resources where you can grant this role:

- Dataset

Masked Reader

(roles/bigquerydatapolicy.maskedReader)

Masked read access to subresources tagged by the policy tag associated with a data policy, for example, BigQuery columns.

Creating custom roles

In IAM, predefined roles have been designed to cover most use cases; therefore, the need for custom roles for BigQuery should be rare and be treated as exceptions.

Custom roles provide access according to a user-specified list of permissions. For example, you may consider creating a custom role when a principal needs a permission, but each predefined role that includes that permission also includes other permissions that the principal does not need and should not have.

Before you create custom roles, ensure that you have a firm understanding of the Google Cloud resource hierarchy and IAM custom roles.

References

- Learn more about predefined IAM roles for BigQuery from the documentation titled "BigQuery predefined IAM roles."
- Learn more about creating and managing custom IAM roles from the documentation titled "Creating and managing custom roles."
- Explore a framework for choosing and implementing predefined IAM roles from the documentation titled "Choose predefined roles."
- Learn how to get the metadata for a specific predefined or custom role to see permissions contained in the role from the documentation titled "Getting the role metadata."
- Learn more about Cloud Audit Logs for BigQuery from the documentation titled "BigQuery audit logs overview."
- Learn more about integrated monitoring, logging, and trace managed services for applications and systems running on Google Cloud including BigQuery from the documentation titled "Google Cloud's operation suite."

Lesson introduction

In this lesson, you will learn about fine-grained access control for BigQuery including how you can leverage data classification and policies at the column and row levels to secure your BigQuery data.

Let's get started!

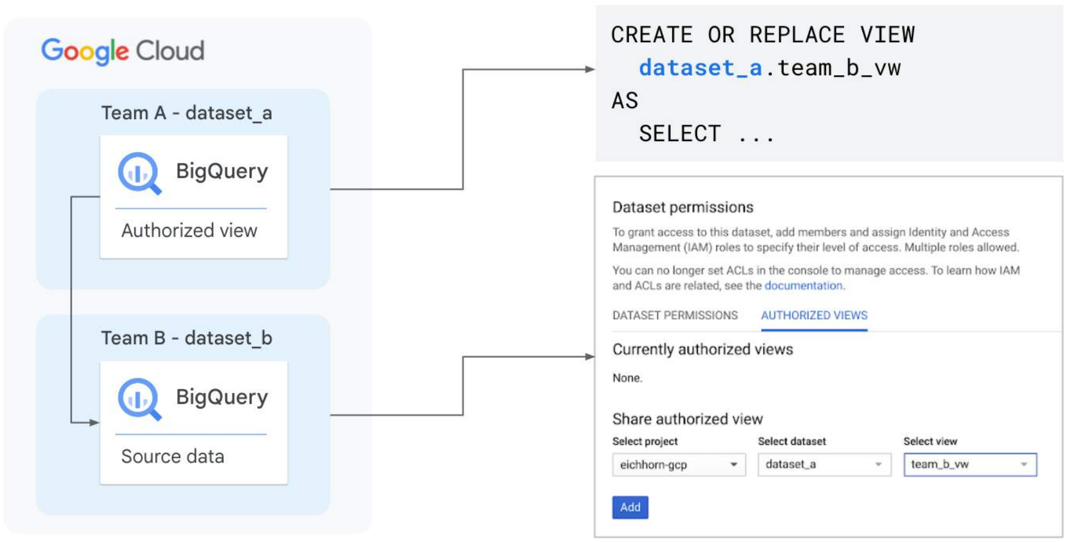
Overview of fine-grained access control

What are your options if you want admins or super users to see all the data in a table, and others to see only a subset of the data?

1. Leverage data classification, column-level permissions, or row-level permissions.
2. Create authorized views of specific tables.
3. Create authorized datasets to authorize all of the views in a specified dataset to access the data in a second dataset.
4. Create authorized functions to share specific query results such as data aggregations without giving access to the underlying data.

Authorized views

An authorized view lets you share query results with particular users and groups without giving them access to the underlying source data. You can also use the view's SQL query to restrict the columns (fields) the users are able to query.



```
CREATE OR REPLACE VIEW
  dataset_a.team_b_vw
AS
  SELECT ...
```

Dataset permissions

To grant access to this dataset, add members and assign Identity and Access Management (IAM) roles to specify their level of access. Multiple roles allowed.

You can no longer set ACLs in the console to manage access. To learn how IAM and ACLs are related, see the documentation.

Currently authorized views

None.

Share authorized view

Select project	Select dataset	Select view
eichhorn-gcp	dataset_a	team_b_vw

Authorized views should be created in a different dataset from the source data. That way, data owners can give users access to the authorized view without simultaneously granting access to the underlying data.

Lesson 16 of 19

Securing data with classification

Data classification is the process of categorizing data into types, forms, or categories by using their distinct characteristics. By classifying your data, you can apply appropriate governance policies to different types of data such as sensitive data and personally identifiable information (PII).

Identify and protect your sensitive data

Data loss prevention

Cloud Data Loss Prevention (DLP) is a fully managed service that lets Google Cloud customers identify, classify, and protect sensitive data at scale. Cloud DLP uses more than 150 predefined detectors to identify patterns, formats, and checksums.

Cloud DLP also provides a set of tools to de-identify your data including masking, tokenization, pseudonymization, date shifting, and more, all without replicating customer data.

You can also create custom detectors using a dictionary or a regular expression. You can add hotword rules to increase the accuracy of findings and set exclusion rules to reduce the number of false positives.

Column-level security

Column-level permissions provide fine-grained access to sensitive columns using policy tags, or type-based classification, of data. Using BigQuery column-level security, you can create policies that check, at query time, whether a user has proper access.

How do you restrict access at the column level? Let's examine the steps.

Click the + icon to expand and learn more.

Step 1: Define taxonomy and policy tags.

Create and manage a taxonomy and policy tags for your data.

An example taxonomy might group all data types into three top-level policy tags depending on the sensitivity of the data: High, Medium, and Low. The High policy tag could contain information such as credit card numbers, while the Low policy tag could contain information such as username.

Step 2: Assign policy tags to your BigQuery columns.

In BigQuery, use schema annotations to assign a policy tag to each column where you want to restrict access.

Step 3: Enforce access control on the taxonomy. –

Enforcing access control causes the access restrictions defined for all of the policy tags in the taxonomy to be applied.

Step 4: Manage access on the policy tags. –

Use IAM policies to restrict access to each policy tag. The policy is in effect for each column that belongs to the policy tag.

Understanding column-level permissions

If the write operation is an *insert*...

...fine-grained read access is not required.
However, the user doesn't have access to read the data that was inserted, unless the user has fine-grained read access.

If the write operation is an *update, delete,* or *merge*...

...the user can't perform the operation unless the user has fine-grained read access on the read columns.

Best practices for column-level access control

Use data profiles to identify data columns that need to be tagged

To determine the types of sensitive data that you have and which columns need policy tags, consider generating profiles about your data across an organization, folder, or project using Cloud Data Loss Prevention. Data profiles contain metrics and metadata about your tables and help you determine where sensitive and high-risk data reside.

Build a hierarchy of data classes

First, consider what kinds of data your organization processes. Usually there are a small number of data classes managed by an organization. For example, an organization could have data classes such as:

- PII data
- Financial data
- Customer order history

A single data class can be applied to multiple data columns using a policy tag. You should leverage this level of abstraction to efficiently manage many columns with only a few policy tags.

Second, consider if there are groups of people who need different access to different data classes. For example, one group needs access to business-sensitive data such as revenues and customer history. Another group needs access to personally identifiable data (PII) like phone numbers and addresses.

Keep in mind that you can group policy tags together in a tree. Sometimes it is helpful to create a root policy tag that contains all of the other policy tags.

Lesson 18 of 19

Row-level security

Row-level security lets you filter data and enable access to specific rows in a table based on qualifying user conditions. Row-level security extends the principle of least privilege by enabling fine-grained access control to a subset of data in a BigQuery table by means of row-level access policies.

Understanding row-level access policies

Row-level security involves the creation of row-level access policies on a target BigQuery table. These policies act as filters to hide or display certain rows of data, depending on whether a user or group is in an allowed list.

When you create a row-level access policy, you specify the table by name, and which users or groups (called the grantee-list) should have access to certain row data. The policy includes the data on which you wish to filter, called the filter_expression. The filter_expression functions like a WHERE clause in a typical query.

Click the + icon to expand and learn more.

How do row-level access policies interact with BigQuery access controls?

Row-level access policies can coexist on a table with column-level security as well as table-level, dataset-level, and project-level access controls. A BigQuery table can also have multiple row-level access policies.

Who can create row-level access policies?

Authorized users with the IAM roles of BigQuery Admin or BigQuery DataOwner can create row-level access policies on a BigQuery table.

Choosing between row-level security and other methods of securing data

Authorized views, row-level access policies, and storing data in separate tables all provide different levels of security, performance, and convenience. Selecting the right mechanism for your use case is important to ensure the proper level of security for your data. Let's examine each one.

Click the tabs to learn more.

AUTHORIZED VIEWS | ROW-LEVEL ACCESS POLICIES | DATA IN SEPARATE TABLES

Recommended for the following:

- Use cases where flexibility and performance are most important.
- For example: sharing data within the same work group.

Possible vulnerabilities:

- Carefully crafted queries, query duration, and other types of side-channel attacks

AUTHORIZED VIEWS | ROW-LEVEL ACCESS POLICIES | DATA IN SEPARATE TABLES

Recommended for the following:

- Use cases where you want all users to query the same table, but you also want some of those users to have access to more data.
- For example: when everyone shares the same dashboard, but some users have access to more data.
- For example: sharing table slices within your organization.

Possible vulnerabilities:

- Carefully crafted queries and query duration side-channel attacks

AUTHORIZED VIEWS | ROW-LEVEL ACCESS POLICIES | DATA IN SEPARATE TABLES

Recommended for the following:

- Use cases in which the total number of rows must be secret.
- For example: sharing data outside your organization, such as with third-party partners and vendors.

Possible vulnerabilities:

- Complete isolation

Best practices for row-level security

Here are some best practices for row-level security in BigQuery.

- Avoid making row access policies that filter on clustered and partitioned columns.
- Restrict user permissions to limit side-channel attacks and data tampering.
- Use row-level security only within organizations, not across organizations.
- Use the BigQuery Filtered Data Viewer role with caution.

References

- Learn more about how to use Cloud DLP to inspect BigQuery data from the documentation titled "Using Cloud DLP to scan BigQuery data."
- Learn more about authorized datasets and views from the documentation titled "Authorized datasets" and "Authorized views."
- Learn more about column-level security from the documentation titled "Introduction to column-level access control."
- Learn more about row-level security from the documentation titled "Introduction to BigQuery row-level security."