



## Working with Audit Logs



In this module, we investigate the core Cloud Audit Logs that Google Cloud collects.

---

# Agenda

Audit Logs

Data Access Logging

Audit Logs Entry Format

Best Practices



We'll start with an overview of what audit logs do for us, then we'll move to using the optional Data Access logs, explore the format of audit log entries, and end up with some logging best practices.

---

# Agenda

## Audit Logs

Data Access Logging

Audit Logs Entry Format

Best Practices



In terms of sheer volume of useful information, probably the most important group of logs in Google Cloud are the audit logs.

## Cloud Audit Logs: “Who Did What, Where, and When?”

Admin Activity	System Event	Data Access
Record modifications to <a href="#">configuration</a> or <a href="#">metadata</a>	Record GCP <a href="#">non-human</a> admin actions that modify <a href="#">configurations</a>	Record calls that read <a href="#">metadata</a> , <a href="#">configurations</a> , or that create, modify, or read <a href="#">user-provided data</a>
Retention is <a href="#">400 days</a>	Retention is <a href="#">400 days</a>	Retention is <a href="#">1-3650 days</a> (30 default)
<a href="#">Immutable</a> and available at <a href="#">no charge</a>	<a href="#">Immutable</a> and available at <a href="#">no charge</a>	
Stored in the <a href="#">_Required</a> log storage bucket	Stored in the <a href="#">_Required</a> log storage bucket	
<a href="#">“Who added that VM?”</a>	<a href="#">“Did a live-migration event occur?”</a>	<a href="#">“Who modified that Cloud Storage file?”</a>
<a href="#">Always enabled</a>	<a href="#">Always enabled</a>	<a href="#">Needs to be enabled</a>



[Cloud Audit Logs](#) help answer the question, “Who did what, when, and where?” It maintains three audit logs for each Google Cloud project, folder, and organization: **Admin Activity**, **Data Access**, and **System Event**.

All Google Cloud services will eventually provide audit logs. For now, see the [Google services with audit logs](#) documentation for coverage details.

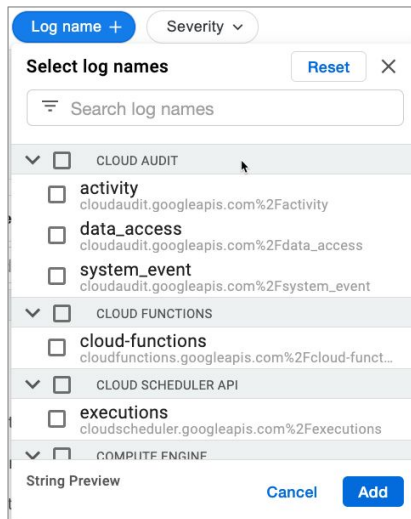
Admin Activity audit logs contain log entries for API calls or other administrative actions that modify the configuration or metadata of resources. For example, these logs record when users create VM instances or change Cloud Identity and Access Management permissions. They are always on, are retained for 400 days, and are available at no charge. To view these logs, you must have the Cloud IAM role **Logging/Logs Viewer** or **Project/Viewer**.

System Event audit logs contain log entries for Google Cloud administrative actions that modify the configuration of resources. System Event audit logs are generated by Google systems; they are not driven by direct user action. They are always enabled, free, retained for 400 days, and to view these logs, you must have the Cloud IAM role **Logging/Logs Viewer** or **Project/Viewer**.

Data Access audit logs contain API calls that read the configuration or metadata of resources, as well as user-driven API calls that create, modify, or read user-provided resource data. Data Access audit logs do not record the data-access operations on resources that are publicly shared (available to **All Users** or **All Authenticated**

**Users**), or that can be accessed without logging into Google Cloud. They are disabled by default (except for BigQuery), and when enabled, the default retention is 30 days. To view these logs, you must have the Cloud IAM roles **Logging/Private Logs Viewer** or **Project/Owner**.

## Filtering audit logs, log names



The Cloud Audit Logs are easy to find using the **Log name** drop-down menu. Note: typing *clouddaudit* into the filter box is frequently quicker than scrolling. If one of the three audit logs is missing, that simply means it doesn't currently have any entries.

To manually specify the log files directly into the query, use:

`logName=("projects/[PROJECT_ID]/logs/cloudaudit.googleapis.com%2Factivity" OR "projects/[PROJECT_ID]/logs/cloudaudit.googleapis.com%2Fdata_access" OR "projects/[PROJECT_ID]/logs/cloudaudit.googleapis.com%2Fsystem_event")`

## Filtering audit logs, an example

```
1 resource.type="gae_app"
2 resource.labels.module_id="default"
3 logName=("projects/devproject-ashok/logs/cloudaudit.googleapis.com%2Fdata_access" OR "projects/devproject-
4 ashok/logs/cloudaudit.googleapis.com%2Factivity")
```

"Control + Space" for autocomplete suggestions ?

Submit Filter

Last 7 days

Jump to now

2017-10-02 PDT

View Options

▶	i	16:11:18.605	App Engine	CreateVersion	default:codelab-default-v000	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.goo...	:
▶	i	16:11:24.219	App Engine	CreateVersion	default:codelab-default-v000	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.goo...	:
▶	i	16:46:39.209	App Engine	CreateVersion	default:codelab-default-v001	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.goo...	:
▶	i	16:46:51.121	App Engine	CreateVersion	default:codelab-default-v001	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.goo...	:
▶	i	16:47:59.231	App Engine	UpdateService	default	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.googleapis.com/google.cl...	:
▶	i	16:48:00.341	App Engine	UpdateService	default	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.googleapis.com/google.cl...	:
▶	i	16:52:25.102	App Engine	UpdateService	default	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.googleapis.com/google.cl...	:
▶	i	16:52:26.394	App Engine	UpdateService	default	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.googleapis.com/google.cl...	:
▶	i	16:55:07.800	App Engine	DeleteVersion	default:codelab-default-v000	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.goo...	:
▶	i	16:55:10.733	App Engine	DeleteVersion	default:codelab-default-v000	949570674432-compute@developer.gserviceaccount.com	{"@type": "type.goo...	:
▶	!!	16:55:31.132	App Engine	UpdateService	default	ashokholla@google.com	{"@type": "type.googleapis.com/google.cloud.audit.AuditLog", "status": "...	:



Here's an example where:

- The resource is set to App Engine.
- The log names are then set to the data access and admin activity logs for a single project.

You see the results below the query.

## Access Transparency logs



Show **how** and **why** customer data is accessed  
once it has been stored in Google Cloud



Whether it's a hardware support engineer, or a rep working on a ticket, having dedicated experts manage parts of the infrastructure is a key benefit of operating in Google Cloud.



## Access Transparency logs



Show **how** and **why** customer data is accessed  
once it has been stored in Google Cloud



Logs of accesses



To Cloud and Apps customer data



By human Googlers



Access Transparency logs help by providing logs of accesses to your data by human Googlers (as opposed to automated systems).

## Access Transparency logs



Show **how** and **why** customer data is accessed  
once it has been stored in Google Cloud



Logs of accesses



To Cloud and Apps customer data



By human Googlers



Provided to enterprises



In near real time



Supports approval and surfaced  
through App APIs and UIs,

**Security Command Center**

Enterprises with appropriate support packages can enable the logs, and receive the log events in near-real time, surfaced through the APIs, Cloud Logging, and Security Command Center.

---

# Agenda

Audit Logs

Data Access Logging

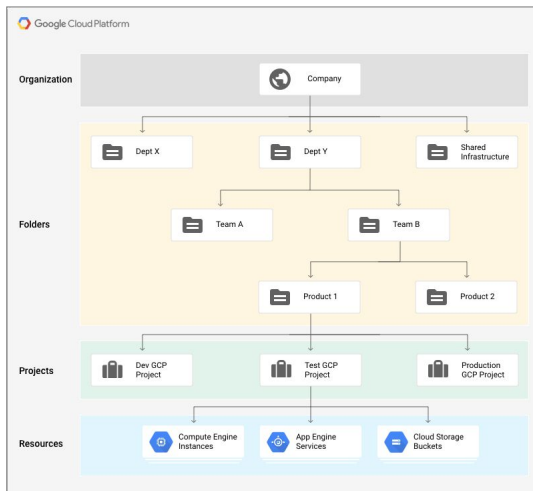
Audit Logs Entry Format

Best Practices



Let's continue by taking a look at Data Access logs.

## Data Access log enablement scope



Google Cloud

- Enable at:
  - Organization
  - Folder
  - Project
  - Resource
- Added cost

Data Access logs can be enabled at the organization, folder, project, or resources levels.

The added logging does add to the cost, currently: \$.50 per gigabyte for ingestion.

# Enabling Data Access logging per Google Cloud service

The screenshot shows the Google Cloud Platform interface with the IAM & Admin section selected. The 'Audit Logs' page is displayed, showing a table of audit logs for various Google Cloud services. The 'Google Cloud Storage' service is highlighted, and its configuration is shown on the right. The configuration panel for Google Cloud Storage shows that 'Data Read' and 'Data Write' are enabled, while 'Admin Read' and 'Admin Write' are disabled. A 'SAVE' button is visible at the bottom of the configuration panel.

Title	Admin Read	Data Read	Data Write	Exemptions
<input type="checkbox"/> Genomics API	--	--	--	0
<input type="checkbox"/> GKE Connect/Hub APIs	--	--	--	0
<input type="checkbox"/> Google App Engine Admin API	--	--	--	0
<input type="checkbox"/> Google Cloud Deployment Manager V2 API	--	--	--	0
<input type="checkbox"/> Google Cloud DNS API	--	--	--	0
<input checked="" type="checkbox"/> Google Cloud Storage	--	--	--	0
<input type="checkbox"/> Google Workspace Add-ons API	--	--	--	0
<input type="checkbox"/> Identity and Access Management (IAM) API	--	--	--	0
<input type="checkbox"/> Identity Toolkit API	--	--	--	0
<input type="checkbox"/> Kubernetes Engine API	--	--	--	0
<input type="checkbox"/> Pub/Sub Lite API	--	--	--	0
<input type="checkbox"/> reCAPTCHA Enterprise API	--	--	--	0
<input type="checkbox"/> Recommendations AI API	--	--	--	0
<input type="checkbox"/> Secret Manager API	--	--	--	0

**Google Cloud Storage**

LOG TYPE EXEMPTED USERS

Turn on/off audit logging for selected services.

☐ Admin Read  
☒ Admin Write  
☒ Data Read  
☒ Data Write

**SAVE**



Data Access logs are disabled by default, for everything but BigQuery. They may be enabled and configured at the organization, folder, project, or service level.

# Enabling Data Access logging per Google Cloud service

The screenshot shows the Google Cloud Platform console with the IAM & Admin section selected. The 'Audit Logs' page is displayed, showing a table of services and their audit log configurations. The 'Google Cloud Storage' service is highlighted, and a modal window is open to configure its audit logging.

Title	Admin Read	Data Read	Data Write	Exemptions
Genomics API	—	—	—	0
GKE Connect/Hub APIs	—	—	—	0
Google App Engine Admin API	—	—	—	0
Google Cloud Deployment Manager V2 API	—	—	—	0
Google Cloud DNS API	—	—	—	0
Google Cloud Storage	—	—	—	0
Google Workspace Add-ons API	—	—	—	0
Identity and Access Management (IAM) API	—	—	—	0
Identity Toolkit API	—	—	—	0
Kubernetes Engine API	—	—	—	0
Pub/Sub Lite API	—	—	—	0
reCAPTCHA Enterprise API	—	—	—	0
Recommendations AI API	—	—	—	0
Secret Manager API	—	—	—	0

**Google Cloud Storage**

LOG TYPE | EXEMPTED USERS

Turn on/off audit logging for selected services.

- ☐ Admin Read
- ☒ Admin Write
- ☒ Data Read
- ☒ Data Write

**SAVE**



You can control what type of information is kept in the audit logs.

There are three types of Data Access audit log information:

- **Admin-read:** Records operations that read metadata or configuration information. For example, you looked at the configurations for your bucket.
- **Data-read:** Records operations that read user-provided data. For example, you listed files and then downloaded one from GCS.
- **Data-write:** Records operations that write user-provided data. For example, you created a new GCS file.

# Enabling Data Access logging per Google Cloud service

The screenshot displays the Google Cloud Platform interface, specifically the IAM & Admin section. The left sidebar shows the navigation menu with 'Audit Logs' selected. The main content area is titled 'Audit Logs' and 'DEFAULT AUDIT CONFIG'. It features a table with columns for 'Title', 'Admin Read', 'Data Read', 'Data Write', and 'Exemptions'. The 'Google Cloud Storage' entry is highlighted, and its 'Data Write' checkbox is checked. A modal window titled 'Google Cloud Storage' is open, showing the 'EXEMPTED USERS' tab. This modal includes a 'Turn on/off audit logging' section with checkboxes for 'Admin Read', 'Admin Write', 'Data Read', and 'Data Write'. The 'Data Write' checkbox is checked. Below this, there is a section for 'Exempted users' with a text input field labeled 'New user' and a 'SAVE' button. The modal also includes a section for 'Exempt Log Types' with checkboxes for 'Admin Read', 'Data Read', and 'Data Write'.

Title	Admin Read	Data Read	Data Write	Exemptions
Genomics API	—	—	—	0
GKE Connect/Hub APIs	—	—	—	0
Google App Engine Admin API	—	—	—	0
Google Cloud Deployment Manager V2 API	—	—	—	0
Google Cloud DNS API	—	—	—	0
Google Cloud Storage	—	—	—	0
Google Workspace Add-ons API	—	—	—	0
Identity and Access Management (IAM) API	—	—	—	0
Identity Toolkit API	—	—	—	0
Kubernetes Engine API	—	—	—	0
Pub/Sub Lite API	—	—	—	0
reCAPTCHA Enterprise API	—	—	—	0
Recommendations AI API	—	—	—	0
Secret Manager API	—	—	—	0

You can exempt specific users or groups from having their data accesses recorded.

As an example, you might decide to exempt your internal testing accounts from having their Cloud Debugger operations recorded.

## Setting the default Data Access logging behavior

[←](#) Default audit config

### Default audit configuration

Set a default audit logging configuration for all Google Cloud Platform services. Default configurations set at the organization level are inherited to all projects in that organization.

[LOG TYPE](#)[EXEMPTED USERS](#)

Turn on/off audit logging for selected services.

☐ Admin Read

☒ Admin Write

☐ Data Read

☐ Data Write

[SAVE](#)



Set a configuration for all new and existing Google Cloud services in your project, folder, or organization inheritance, ensuring that Data Access audit logs are captured.



## Programmatically enabling Data Access logging

```
auditConfigs:
- auditLogConfigs:
  - logType: ADMIN_READ
  - logType: DATA_READ
  - logType: DATA_WRITE
  service: run.googleapis.com #Could also be allServices
bindings:
- members:
  ...
```



You can also use `gcloud` or the API to enable Data Access logging.

## Programmatically enabling Data Access logging

```
auditConfigs:
- auditLogConfigs:
  - logType: ADMIN_READ
  - logType: DATA_READ
  - logType: DATA_WRITE
  service: run.googleapis.com #Could also be allServices
bindings:
- members:
  ...
```

- `gcloud projects get-iam-policy [project-id] > policy.yaml`



If you're using gcloud, frequently, the easiest way is to get the current IAM policies, as seen in the bullet, and write them to a file.

## Programmatically enabling Data Access logging

```
auditConfigs:
- auditLogConfigs:
  - logType: ADMIN_READ
  - logType: DATA_READ
  - logType: DATA_WRITE
  service: run.googleapis.com #Could also be allServices
bindings:
- members:
  ...
```

- `gcloud projects get-iam-policy [project-id] > policy.yaml`
- Add/edit the auditLogConfigs



Then you can edit the file to add or edit the auditLogConfigs.

You can also add the log details per service, like this example is enabling logging for Cloud Run, or even enable logging on all services.

## Programmatically enabling Data Access logging

```
auditConfigs:
- auditLogConfigs:
  - logType: ADMIN_READ
  - logType: DATA_READ
  - logType: DATA_WRITE
  service: run.googleapis.com #Could also be allServices
bindings:
- members:
  ...
```

- `gcloud projects get-iam-policy [project-id] > policy.yaml`
- Add/edit the auditLogConfigs
- `gcloud projects set-iam-policy [project-id] policy.yaml`



Then, as seen in the bullet, you would set that as the new IAM policy.

---

# Agenda

Audit Logs

Data Access Logging

**Audit Logs Entry Format**

Best Practices



Now that we can enable the logs we need, let's examine the logging entries themselves.

## Audit Log entries

```
{
  insertId: "-77e5fge38tyo"
  logName: "projects/patrick-haggerty/logs/cloudaudit.googleapis.com%2Fdata_access"
  operation: {
    first: true
    id: "1581200795118-patrick-haggerty:bquxjob_56996f5_17026e67aa2"
    producer: "bigquery.googleapis.com"
  }
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {
      principalEmail: "patrick.haggerty@roittraining.com"
    }
  }
}
```



Every audit log entry in Cloud Logging is an object of type [LogEntry](#).

What distinguishes an audit log entry from other log entries is the `protoPayload` field, which contains an [AuditLog](#) object that stores the audit logging data.

## Audit Log entries

```
{
  insertId: "-77e5fge38tvo"
  logName: "projects/patrick-haggerty/logs/cloudaudit.googleapis.com%2Fdata_access"
  operation: {
    first: true
    id: "1581200795118-patrick-haggerty:bquxjob_56996f5_17026e67aa2"
    producer: "bigquery.googleapis.com"
  }
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {
      principalEmail: "patrick.haggerty@roittraining.com"
    }
  }
}
```



Here, note the log name. This tells us that we're looking at an example from the Data Access log.

## Audit Log entries

```
{
  insertId: "-77e5fge38tyo"
  logName: "projects/patrick-haggerty/logs/cloudaudit.googleapis.com%2Fdata_access"
  operation: {
    first: true
    id: "1581200795118-patrick-haggerty:bquxjob_56996f5_17026e67aa2"
    producer: "bigquery.googleapis.com"
  }
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {
      principalEmail: "patrick.haggerty@roittraining.com"
    }
  }
}
```



Now, note the AuditLog type in the protoPayload.



## Audit Log entries

```
authorizationInfo: [2]
methodName: "jobservice.getqueryresults"
requestMetadata: {...}
resourceName: "projects/patrick-haggerty/queries/bqjob_1eb1f384_17026e9d185"
serviceData: {...}
serviceName: "bigquery.googleapis.com"
status: {...} }
receiveTimestamp: "2020-02-08T22:27:06.410866127Z"
resource: {
  labels: { project_id: "patrick-haggerty", op_unit: "USA"
  }
  type: "bigquery_resource"
}
severity: "INFO"
timestamp: "2020-02-08T22:27:05.908Z"
}
```



Google has a standard [List of official service names](#). You can use this list as a handy reference.

## Audit Log entries

```
authorizationInfo: [2]
methodName: "jobservice.getqueryresults"
requestMetadata: {...}
resourceName: "projects/patrick-haggerty/queries/bqjob_1eb1f384_17026e9d185"
serviceData: {...}
serviceName: "bigquery.googleapis.com"
status: {...} }
receiveTimestamp: "2020-02-08T22:27:06.410866127Z"
resource: {
  labels: { project_id: "patrick-haggerty", op_unit: "USA"
  }
  type: "bigquery_resource"
}
severity: "INFO"
timestamp: "2020-02-08T22:27:05.908Z"
}
```

Drill down and the query itself is in here



On this slide, you can tell we're looking at a query that was run in BigQuery.

If you expanded the serviceData field, you could actually see the query itself.

So, when someone at your organization runs that unexpected, \$40,000 query, you can figure out who ran it and what the query was.

Then you can go learn more about price controls and BigQuery.

---

# Agenda

Audit Logs

Data Access Logging

Audit Logs Entry Format

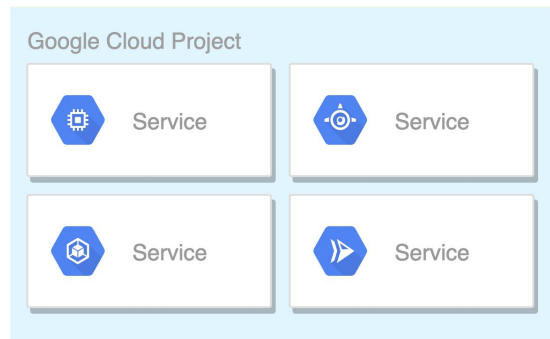
Best Practices



Let's go over a few best practices before we wrap up this module.

## Plan and create test project

- Create a plan for Data Access logging
  - Think Org-wide, then folder, then project
- Create a test project and test plan there
- Roll out



Plan, Plan, Plan.

Like anything in the cloud, start by planning first.

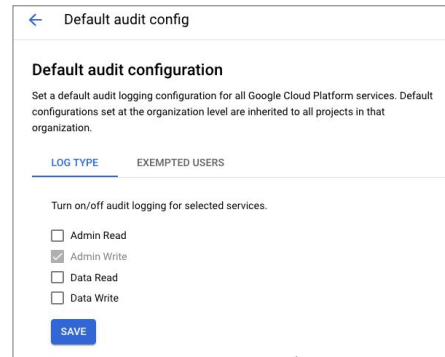
Spend time and create a solid plan for Data Access logs. Think organization, folder, then project. Like most organizations, some of your projects will be very specialized, but usually, they do break down into common organizational types.

Then, create a test project and experiment to see if the logging works the way you expect.

Then roll out, and don't forget automation (Infrastructure as Code, coming soon).

## Decide and set org level data access

- Pro: detailed information on exactly who, accessed/edited/deleted what, and when
  - Free tier
  - Some logs always free
- Con: logs can be quite large
  - \$0.50/GiB



The screenshot shows the 'Default audit config' page in the Google Cloud console. It has a back arrow and the title 'Default audit config'. Below the title is the section 'Default audit configuration' with a description: 'Set a default audit logging configuration for all Google Cloud Platform services. Default configurations set at the organization level are inherited to all projects in that organization.' There are two tabs: 'LOG TYPE' (selected) and 'EXEMPTED USERS'. Under the 'LOG TYPE' tab, it says 'Turn on/off audit logging for selected services.' and lists four checkboxes: 'Admin Read' (unchecked), 'Admin Write' (checked), 'Data Read' (unchecked), and 'Data Write' (unchecked). A blue 'SAVE' button is at the bottom.



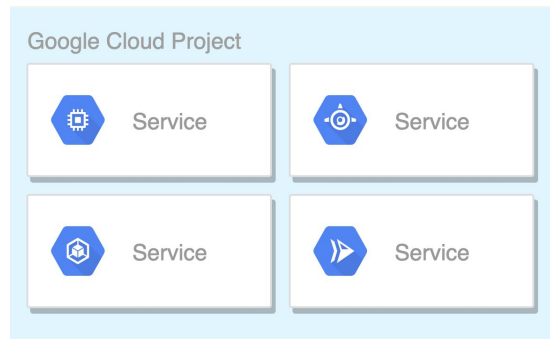
Remember that Data Access logs can be enabled as high as the organization.

The pro would be a lot of very detailed information on exactly who accessed, edited, and deleted what, and when.

The con is that Data Access logs can grow to be quite large, and are billed at \$.50 a gig.

## Standard project logging plans

- Past default logging, what logging requirements exist at the project level
  - Different major project types



Past whatever default logging you institute, every project needs to be analyzed for specialized logging requirements.

## Infrastructure as Code (IaC)



### Deployment Manager

Fully-hosted product, with Google Support  
Closed source, GCP native  
State stored and supported by Google  
Audit logging of all operations



Infrastructure as Code (IaC) is essentially the process of automating the creation and modifications to your infrastructure using a platform that supports configuration files, which can be put through a CI/CD pipeline, like with code.

Let's look at some of your options for implementing this.

Deployment Manager is Google Cloud's native IaC technology. It is fully hosted, has Google support, is auditable, and its state is stored in Google Cloud.

## Infrastructure as Code (IaC)



### Deployment Manager

Fully-hosted product, with Google Support  
Closed source, GCP native  
State stored and supported by Google  
Audit logging of all operations



### Terraform: OSS or Enterprise

Run CLI or pay for enterprise version  
Open source, multi-cloud  
State stored locally or in GCS  
Logging at APIs



Terraform is an open-source package from HashiCorp.

It isn't hosted directly in Google Cloud, though it is installed by default in Cloud Shell.

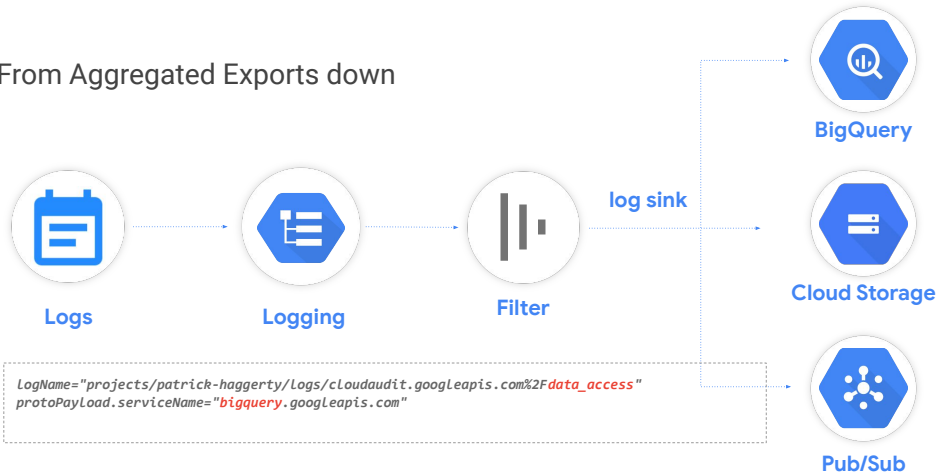
State management is a decision point for your organization.

Options include using Cloud Storage, using a Git repo like GitHub, setting up something local to your organization, or using HashiCorp's pay Enterprise service.



## Plan and configure exports

- From Aggregated Exports down



We've discussed the options and benefits of exporting logs. Again, make this part of your plan.

Start by deciding what, if anything, you will export from Aggregated Exports at the organization level.

Next, decide what options you will use, project by project, folder by folder, etc.

Then, carefully consider your filters—both what they leave in, and what they leave out.

This applies to all logging, not just to exports.

Lastly, carefully consider what, if anything, you will fully exclude from logging.

Remember that **excluded entries will be gone forever.**

## Principle of least privilege

- Side-channel leakage of data through logs is a common issue
- Plan the project to monitoring project relationships
- Use appropriate IAM controls on both Google Cloud-based and exported logs
- Data Access logs contain Personally Identifiable Information (PII)



Side-channel leakage of data through logs is a common issue.

You need to be careful who gets what kind of access, to which logs.

Remember some of the discussions earlier in this course on monitoring workspaces and how a monitoring workspace can monitor the current project, or it can also monitor up to 100 other projects?

That's where your security starts.

Are you monitoring project by project, or are you selectively grouping work projects into higher-level monitored projects?

Use appropriate IAM controls on both Google Cloud-based and exported logs, only allowing the minimal access required to get the job done.

Especially scrutinize the Data Access log permissions, as they will often contain Personally Identifiable Information (PII).

## Scenario: operational monitoring

- CTO: **resourcemanager.organizationAdmin**
  - Assigns permissions to security team and service account
- Security team: **logging.viewer**
  - Ability to view Admin Activity logs
- Security team: **logging.privateLogViewer**
  - Ability to view Data Access logs
- All permissions assigned at Org level
- Control exported data access through Cloud Storage and BigQuery IAM roles
- Explore using Cloud DLP (Data Loss Prevention) to redact PII



Lastly, [a few access scenarios](#), starting with operational monitoring.

These are your high-level teams and assignments.

By job:

CTO: `resourcemanager.organizationAdmin`, so he/she can assign permissions to the security team and service accounts.

The CTO can then give the security team `logging.viewer` so they can view the Admin Activity logs, and `logging.privateLogViewer`, so they can view the Data Access logs.

These permissions are assigned at the organization level, so they are global.

Access control to data exported to Cloud Storage or BigQuery will be secured selectively with IAM.

You might also want to explore Cloud DLP to redact the PII.

## Scenario: Dev teams monitoring Audit Logs

- Security team, same:
  - **logging.viewer**, **logging.privateLogViewer**
- Dev team: **logging.viewer** at folder level
  - See Admin Activity by dev projects in folder
- Dev team: **logging.privateLogViewer** at folder
  - See Data Access logs
- Again, use Cloud Storage or BigQuery IAM to control access to exported logs
  - Providing a Dashboard might be helpful



Now let's move on to development teams.

The security team is unchanged from the last slide. They already have **logging.viewer**, and **logging.privateLogViewer** from the global assignment.

The dev team might get **logging.viewer** at the folder level so they can see the Admin Activity logs for the projects under their development control.

They probably also need **logging.privateLogViewer** at the dev folder so they can see the Data Access logs. Limit data they test with though, so they aren't viewing actual customer information.

Again, use Cloud Storage or BigQuery IAM to control access to exported logs. Prebuilding dashboards might also be a good option.

---

## Scenario: External Auditors

- Provide Dashboards for auditor usage
- **logging.viewer** at Org level
  - See Admin Activity by dev projects in folder
- **bigquery.dataViewer** at exported dataset
  - Backend for Dashboards
- For Cloud Storage, use IAM and/or, signed, temporary, URLs



For external auditors, provide pre-created dashboards where possible.

If they need broad access, you might make them `logging.viewer` at the org level.

For BigQuery, they could be `bigquery.dataViewer` on the exported dataset.

For Cloud Storage, again, you could use IAM, but also remember the temporary access URLs that Cloud Storage supports.

# Lab Intro

## Cloud Audit Logs



In this lab, you investigate Cloud Audit Logging. Cloud Audit Logging maintains multiple audit logs for each project, folder, and organization: Admin Activity tracks changes to Google Cloud configurations and metadata. Data Access is more directly related to who accessed what data, and when.

---

## Quiz

Why are the Data Access logs off by default?

- A. Can be very large
- B. Can be expensive to store
- C. May contain sensitive information
- D. All of the above

---

## Quiz

Why are the Data Access logs off by default?

- A. Can be very large
- B. Can be expensive to store
- C. May contain sensitive information
- D. All of the above



---

## Quiz

If you want to provide an external auditor access to your logs, what IAM role would be best?

- A. Project Viewer
- B. Project Editor
- C. Logging Viewer
- D. Logging Admin

---

## Quiz

If you want to provide an external auditor access to your logs, what IAM role would be best?

- A. Project Viewer
- B. Project Editor
- C. Logging Viewer
- D. Logging Admin

---

## Learned how to...

- Use Admin Activity, Data Access, and System Event audit logs to track who, did what, and when



Very good. After this module, you now know how to:

- Use Cloud Audit Logs: Admin Activity, Data Access, and System Event, to answer the question, "Who, did what, and when?"

Fantastic job.

