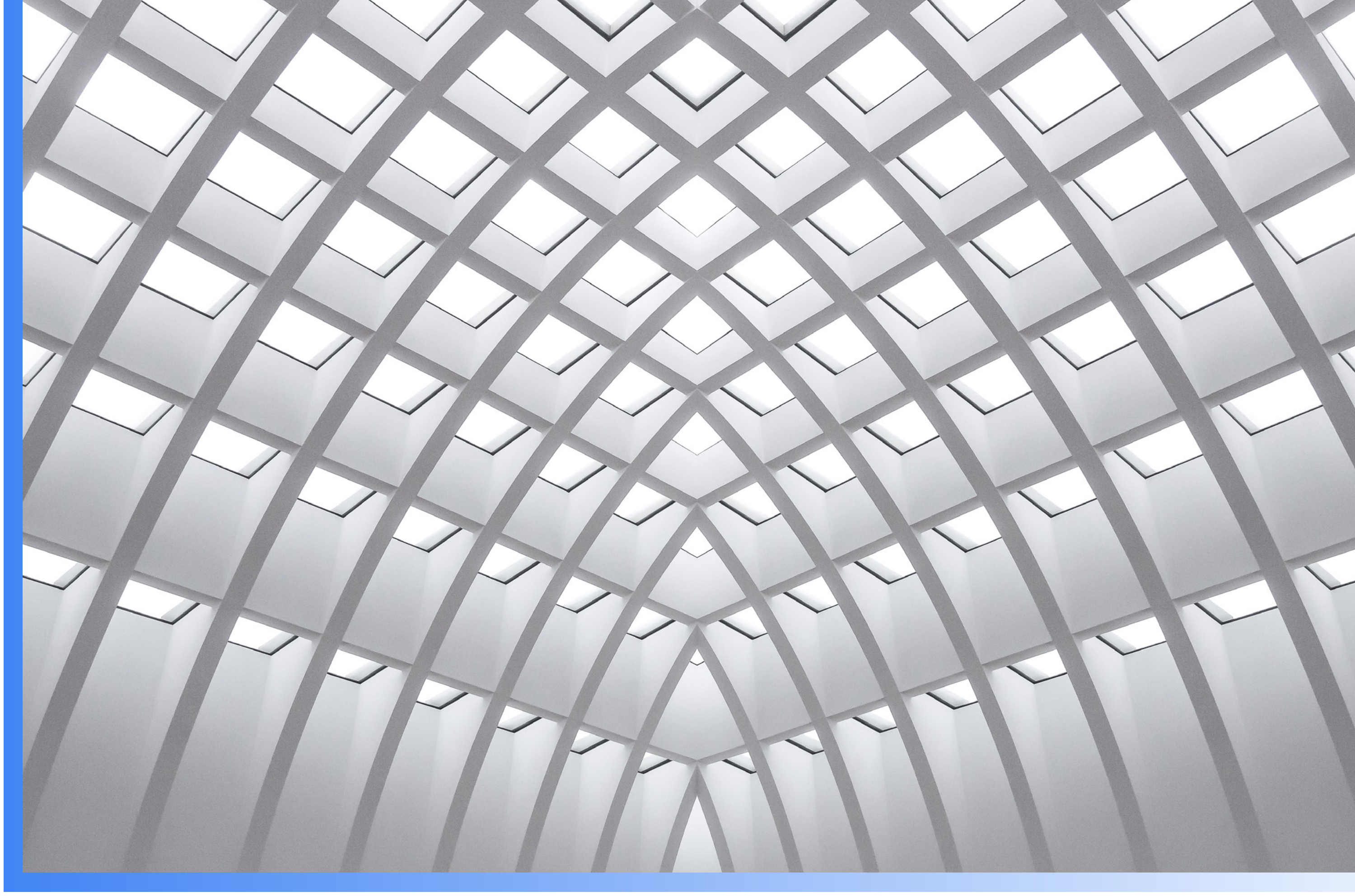


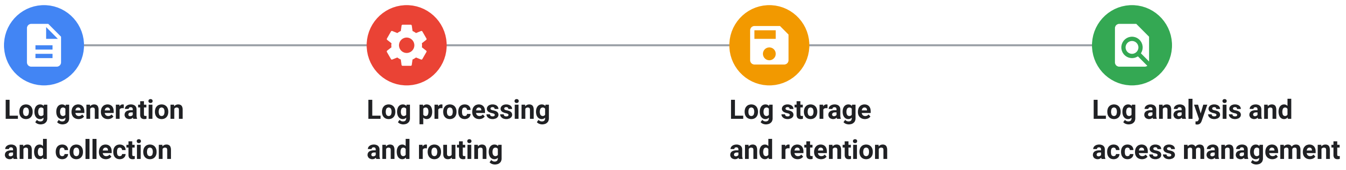
Logs data: A step by step guide for overcoming common compliance challenges

Four solution guides with customer examples

Google Cloud

About this whitepaper

This whitepaper is designed to provide cloud architects with step by step guidance for how to address four of the most common data governance challenges for logs data in Google Cloud. In our companion whitepaper, Data governance: Principles for securing and managing logs, we covered the four-stage lifecycle of logs data and provided best practices for how to configure your systems.


Log generation and collection
Log processing and routing
Log storage and retention
Log analysis and access management

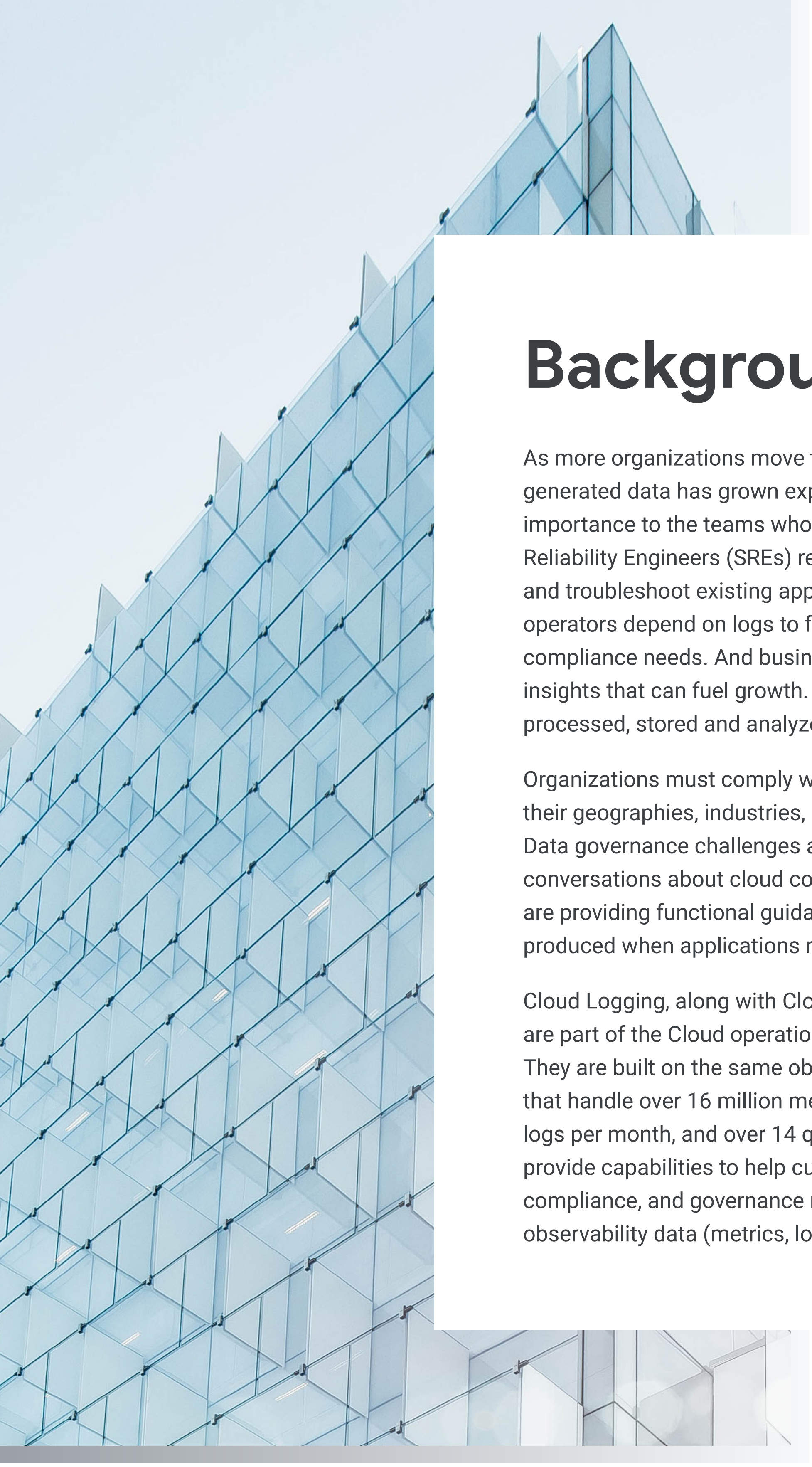
This guide lays out how to implement the strategies discussed in the companion whitepaper with real-world examples from our customers. See the table below for the four data governance challenges, and the solution that this guide will walk you through.

Data Governance requirement	Cloud Logging solution	Target audience
Data residency (e.g. GDPR or FedRAMP)	Store logs in a regional bucket	Cloud architect or Cloud admin (There should be no impact on developers or IT operators, provided they have appropriate access to view logs.)
Access to search across all logs in organization (e.g. compliance audit)	Aggregated sink at org level	
Long term retention of logs (e.g PCI DSS)	Configurable retention per log bucket	
Granular access control (e.g. role based access control restricting access to HIPAA data)	Row level access via log views; field level access via field restrictions on log buckets	

Four of the most common scenarios for logs data governance

Limitations around logs data governance

Please note that saved, shared, and recent queries, dynamically generated short links for sharing links from the cloud console, and log based metrics are not yet covered by Cloud Logging data residency guarantees. Avoid using these features to store sensitive data. Error Reporting, a tool which automatically detects anomalies in log entries, does not yet support data residency guarantees so will not work on logs that are stored in a regional rather than a global bucket. We are invested in your success. The information and best practices in this guide are meant to support you in building solutions that would be used as part of your comprehensive data governance strategy. It is your responsibility to ensure your operations are compliant with any applicable laws and compliance regulations. We plan to keep doing our part by building solutions that help you get there.



Background

As more organizations move to the cloud, the volume of machine generated data has grown exponentially, both in volume and importance to the teams who rely on it. Software engineers and Site Reliability Engineers (SREs) rely on logs to develop new applications and troubleshoot existing apps to meet reliability targets. Security operators depend on logs to find and address threats and meet compliance needs. And business teams leverage logs for invaluable insights that can fuel growth. But first logs must be collected, processed, stored and analyzed.

Organizations must comply with a host of data regulations based on their geographies, industries, and internal governance requirements. Data governance challenges are usually most prominent in conversations about cloud compute and storage. In this guide we are providing functional guidance for what to do with the logs data produced when applications run on that compute infrastructure.

Cloud Logging, along with Cloud Monitoring and several other tools, are part of the Cloud operations suite of managed service offerings. They are built on the same observability platforms used by all of Google that handle over 16 million metrics queries per second, 2.5 exabytes of logs per month, and over 14 quadrillion metric points on disk. They also provide capabilities to help customers achieve a variety of regulatory, compliance, and governance requirements with respect to their observability data (metrics, logs and traces).

Solving common compliance challenges

Below are the four step-by-step solution guides for configuring your system to address the most common compliance challenges for logs data.

Google Cloud

01

Compliance challenge 1:
Configuring a project to store logs in a specific region

Customer example:

A large European retailer needed to store their logs from a project, `InventoryApp1`, in Europe for compliance purposes.

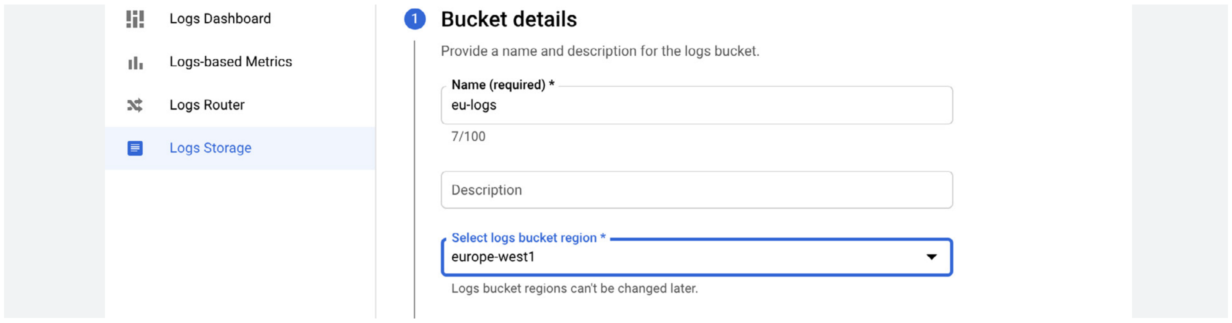
Solution:

They set up Logging to store logs in a specific region, `europe-west1`, in three easy steps.

01

Create a new log bucket in the desired region — In this example, the new bucket will be named `eu-logs` and it will be created in `europe-west1`.

- Go to the Cloud Console > Logging > Logs Storage and click on "Create logs bucket".
- Name the log bucket and choose the desired region. **The region cannot be changed later.** Click Create Bucket.



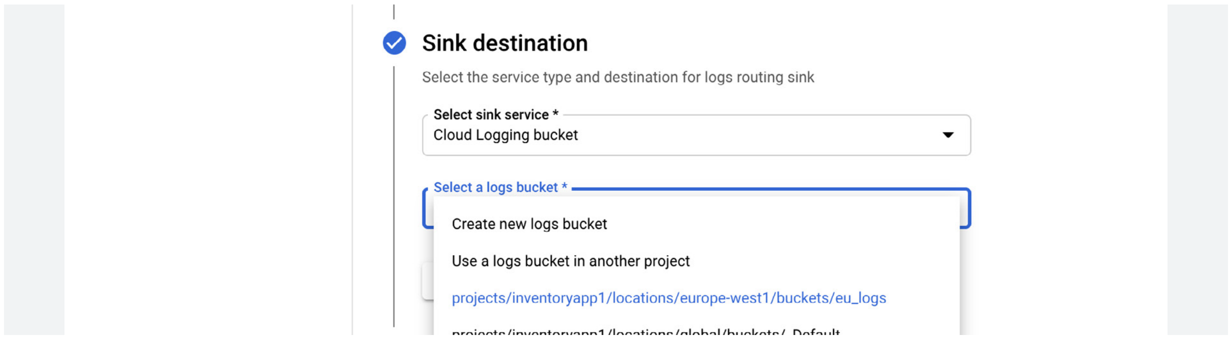
Bucket details
Provide a name and description for the logs bucket.
Name (required) *
eu-logs
7/100
Description
Select logs bucket region *
europe-west1
Logs bucket regions can't be changed later.

02

Point all incoming new logs to the new log storage bucket — Now that you have a new bucket, stop logs from going to the global `_Default` bucket and send them instead to the new `eu-logs` bucket by editing the destination of the existing `_Default` sink to point to the new bucket.

Go to the Logs Router section of the Cloud Console and click on the dots to the right of the `_Default` sink. Select "Edit Sink".

- Under Sink Destination, change the log bucket selected from "projects/.../_Default" to "projects/.../eu_logs". Scroll to the bottom and select "Update sink" to save the changes.


Sink destination
Select the service type and destination for logs routing sink
Select sink service *
Cloud Logging bucket
Select a logs bucket *

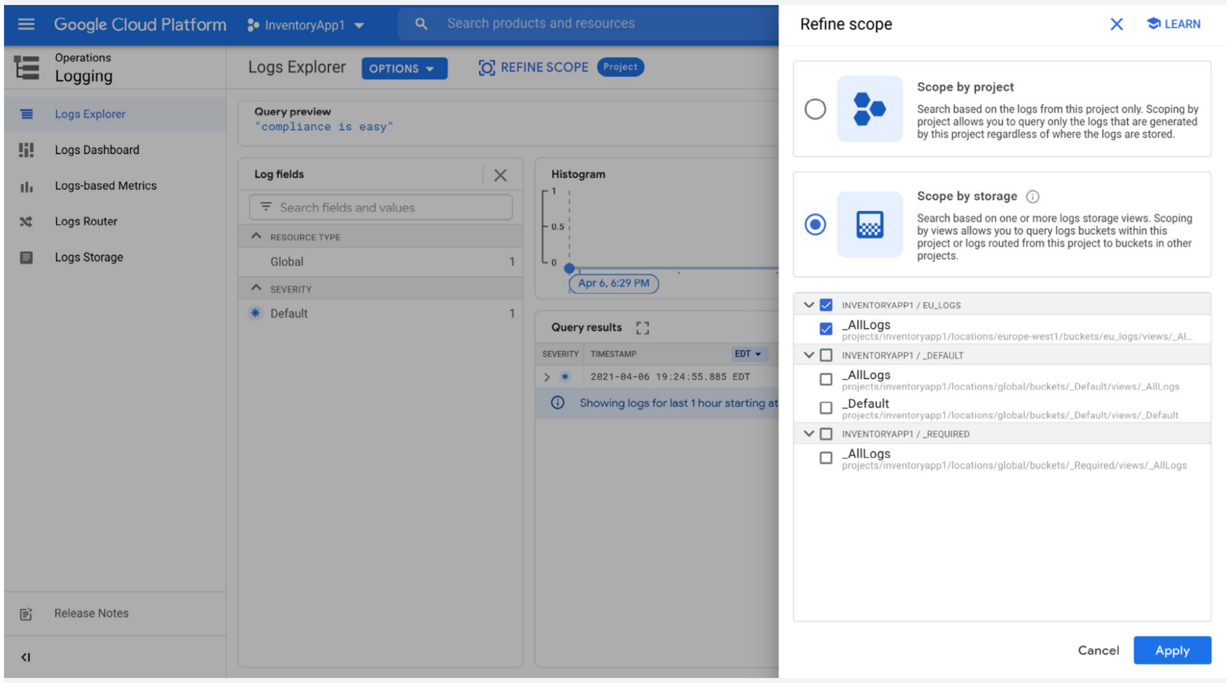
projects/inventoryapp1/locations/europe-west1/buckets/eu_logs
projects/inventoryapp1/locations/global/buckets/_Default

03

Bonus - test to make sure that everything is working correctly — At this point, you're actually done and logs are being stored in the regional bucket. To verify, you can use gcloud to write a log entry to your project and check that the log entry was only in the regional bucket.

- Open Cloud Shell, and use gcloud logging write to send a test log entry with a log name "`eu-test`" and a text message to your project.
  `$ gcloud logging write eu-test "compliance is easy!"`
  You should see confirmation — Created log entry.

- In order to verify that the log entry is saved only in the regional bucket, change the scope from "Project" to "Storage". **Scoping by project allows you to query only the logs that are generated by this project regardless of where the logs are stored. This lets developers find their logs no matter how you organize them for your data governance needs.** Scoping by storage lets you explicitly select one or more log buckets and log views. At the top of the page, to the right of "Logs Explorer", select "Refine Scope (Project)". From the options, select the **`eu-logs`** bucket.



Query preview
"compliance is easy"

RESOURCE TYPE
Global 1
SEVERITY
Default 1

Apr 6, 6:29 PM

SEVERITY TIMESTAMP EDT
2021-04-06 19:24:55.885 EDT
Showing logs for last 1 hour starting at

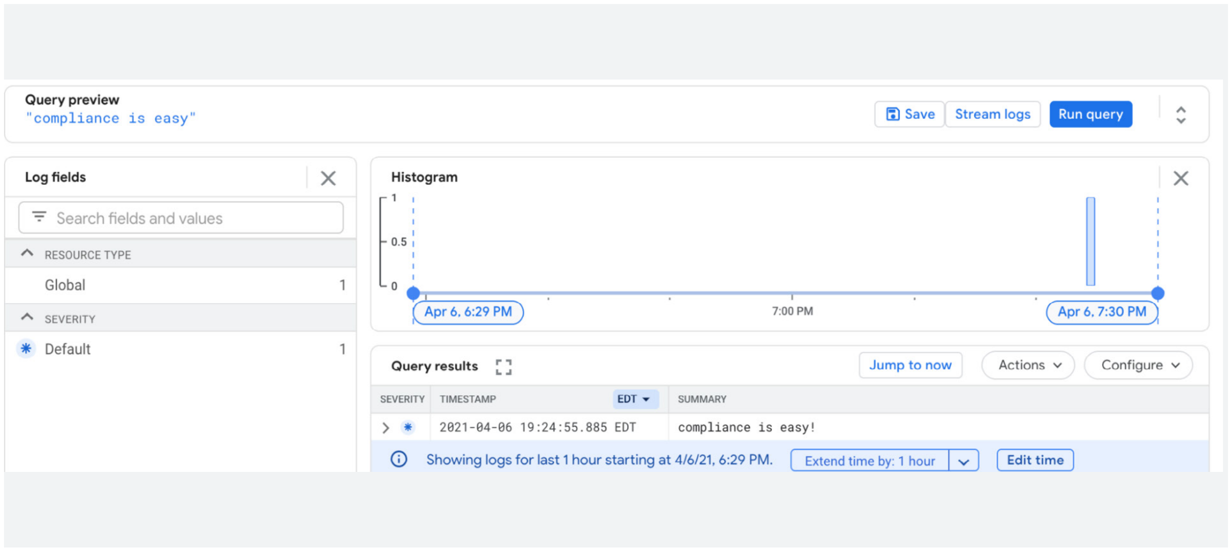
Refine scope

Scope by project
Search based on the logs from this project only. Scoping by project allows you to query only the logs that are generated by this project regardless of where the logs are stored.

[✓ Scope by storage]
Search based on one or more logs storage views. Scoping by views allows you to query logs buckets within this project or logs routed from this project to buckets in other projects.

- [✓] INVENTORYAPP1 / EU_LOGS
  - [✓] _AllLogs projects/inventoryapp1/locations/europe-west1/buckets/eu_logs/views/_Al...
- [ ] INVENTORYAPP1 / _DEFAULT
  - [ ] _AllLogs projects/inventoryapp1/locations/global/buckets/_Default/views/_AllLogs
  - [ ] _Default projects/inventoryapp1/locations/global/buckets/_Default/views/_Default
- [ ] INVENTORYAPP1 / _REQUIRED
  - [ ] _AllLogs projects/inventoryapp1/locations/global/buckets/_Required/views/_AllLogs

- In the query editor, type in the text you entered for your log message, "**`compliance is easy!`**" This does a global search for matching log entries in the selected scope. You should see the matching entry. You can repeat with the **`_Default`** bucket selected to verify that no matching logs are returned.



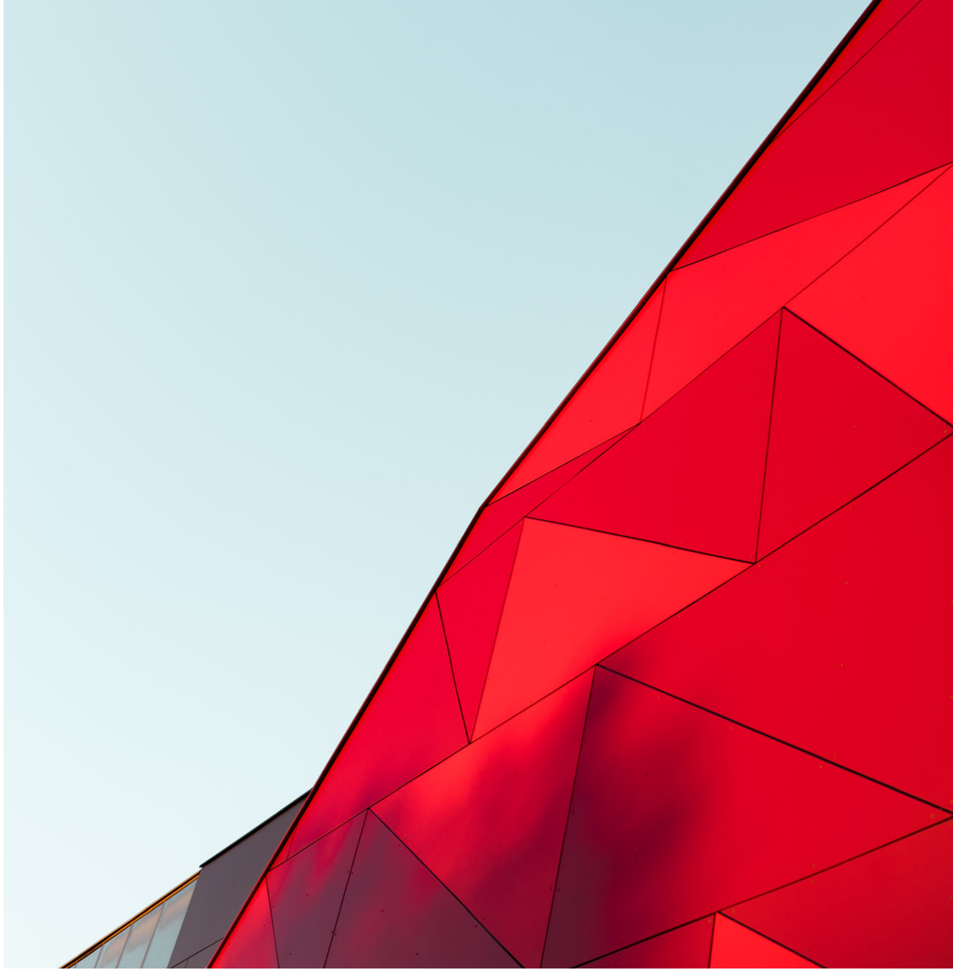
```
"compliance is easy"
```

```
2021-04-06 19:24:55.885 EDT    compliance is easy!
```

Showing logs for last 1 hour starting at 4/6/21, 6:29 PM.

That's it! We've validated that logs are now being stored in `europe-west1`.

Google Cloud



02

Compliance challenge 2:
Centralizing logs with an organization-wide sink

Customer example:

A US based financial institution, needed to store all their audit logs — data access, admin activity, etc. from across their organization in a single project. This enabled them to audit compliance and detect threats, even if the underlying resources and projects were deleted. They also needed to ensure that a malicious project owner could not opt out of sending logs to the centralized project. Lastly they wanted to change existing projects to exclude these logs where possible to avoid extra charges.

Solution:

While they could create a new log sink for every project in their organization and point it to the destination, this would be difficult to manage and has a risk that a malicious actor who compromised a project could disable the log sink on that project. Instead, they created a new log bucket in their region and sent logs from all the child projects to this new log bucket. Recreate the steps they took: together, these logs provide complete audit trails of administrative changes and data accesses of your Google Cloud resources including any access by Google.

01

Create a new log bucket in the desired region to hold your audit logs — As before, create a new bucket, example name: `all-audit-logs-bucket`, in the region of your choice by going to the Logs Storage section of the Cloud Console and clicking on "Create logs bucket." Name the bucket and choose the correct region. Remember that the region cannot be changed later. This new log bucket should be in its own project owned by the security team so they have control over the data.

02

Create an aggregated sink at the organization level using gcloud — Since the Cloud Console doesn't support managing sinks for folders or organizations, use gcloud to create the log sink.

First, list existing sinks at the organization level. Note that this won't recursively list all sinks in all child projects, just sinks that live at the org level. Use this snippet below and replace the organization ID with your organization's ID.

```
$ gcloud logging sinks list --organization=12345
```

You'll see the `_Required` and `_Default` sinks listed, just like for a project.

Next, create a new sink at the org level going to your new project. You can use the command below:

```
$ gcloud logging sinks create all-audit-logs-sink
logging.googleapis.com/projects/PROJECT/locations/LOCATION/buckets/all-
audit-logs-bucket
--log-filter='logName:cloudaudit.googleapis.com'
--description="All audit logs from my org log sink"
--organization=12345   --include-children
```

Breaking this into pieces:

- The destination for the log bucket, logging.googleapis.com/projects/PROJECT/locations/LOCATION/buckets/all-audit-logs-bucket, needs to be modified to include the project name and chosen location of the new bucket.
- For the log filter, there are many ways to capture the relevant logs.
  This uses the substring operator "`:`" on the logName so you don't need to worry about stripping out the project name with a regular expression, which would also work. Our query library has many other example queries if your use case is different.

- The flag `--include-children` tells the log router to evaluate this for all child folders and projects. Otherwise, it would only match audit logs written against the organization directly such as when a new project is created.

03

Grant the new org sink permission to write the logs to the new bucket — This sink will immediately begin trying to send logs to the new bucket but will fail until you grant the service account permission. First, get the service account for the new sink using the gcloud describe command:

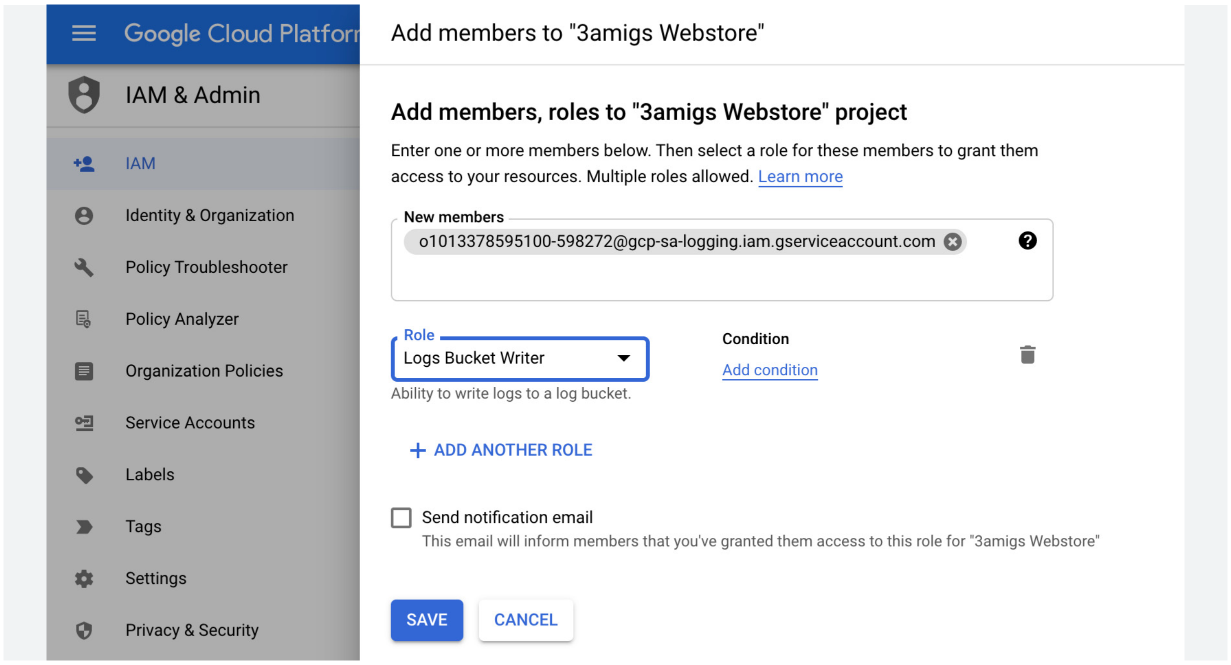
```
$ gcloud logging sinks describe all-audit-logs-sink
--organization=12345
```

You should see a response that includes a writerIdentity:

```
writerIdentity:
serviceAccount:o12345-9876@gcp-sa-logging.iam.gserviceaccount.com
```

Copy this `writerIdentity` and grant it the Logs Bucket Writer permission in Cloud IAM as in the example below (you don't need to include "`serviceAccount:`")

You should now see all audit logs from your entire organization going into the new log bucket.



Add members to "3amigs Webstore"

Add members, roles to "3amigs Webstore" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. Learn more

New members: o1013378595100-598272@gcp-sa-logging.iam.gserviceaccount.com

Role: Logs Bucket Writer

Ability to write logs to a log bucket.

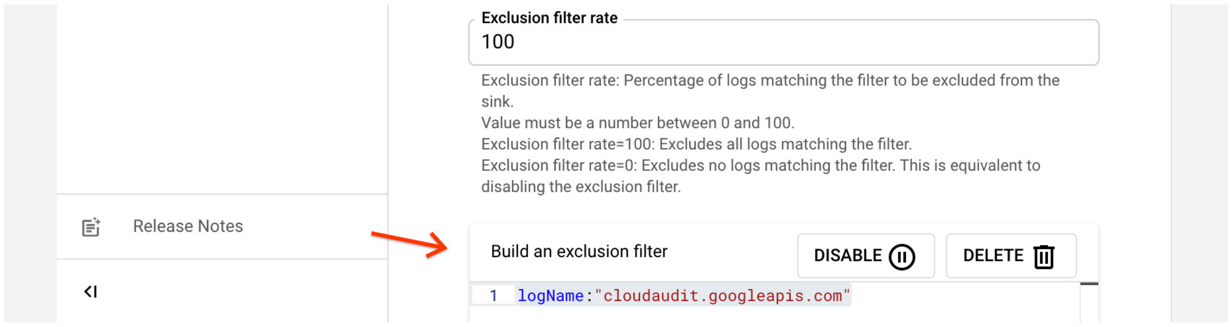
Condition: Add condition

☐ Send notification email

This email will inform members that you've granted them access to this role for "3amigs Webstore"

04

Exclude the same filter from the `_Default` sink. Use the Cloud Console to go to the Logs Router page to edit the `_Default` sink. Add an exclusion filter for the same filter you used for your org sink '`logName:"cloudaudit.googleapis.com`"'. This needs to be repeated for each project that wants to avoid including the matching logs. If you have many projects, you may want to use Terraform or a similar tool to automate the process.

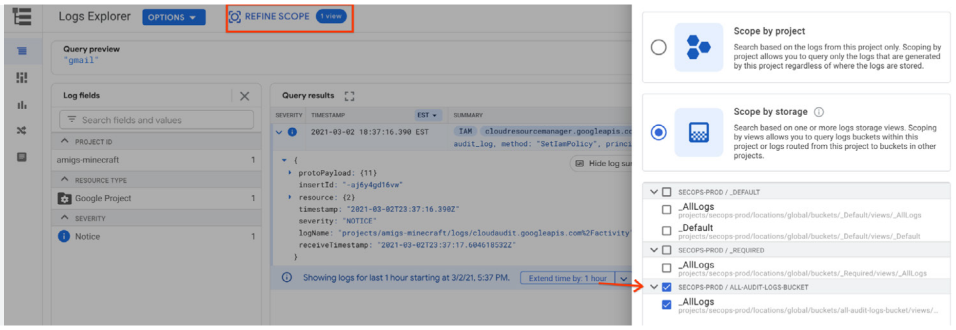

Exclusion filter rate
100
Exclusion filter rate: Percentage of logs matching the filter to be excluded from the sink.
Value must be a number between 0 and 100.
Exclusion filter rate=100: Excludes all logs matching the filter.
Exclusion filter rate=0: Excludes no logs matching the filter. This is equivalent to disabling the exclusion filter.

Build an exclusion filter

```
logName:"cloudaudit.googleapis.com"
```

05

Verify you see the matching logs in Logs Explorer. The Logs Explorer is designed to show only the logs *from* the project you're currently working in. In order to see all the logs from across the organization, simply switch from "Project scope" to "Storage scope" and verify you see the audit logs from across your organization.


Query preview
"gmail"

Log fields
PROJECT ID: amigs-minecraft 1
RESOURCE TYPE: Google Project 1
SEVERITY: Notice 1

Query results

```
SEVERITY  TIMESTAMP                    SUMMARY
          2021-03-02 18:37:16.390 EST  IAM  cloudresourcemanager.googleapis.c
                                       audit_log, method: "SetIamPolicy", princi
{
  protoPayload: {11}
  insertId: "-aj6y4gd16vw"
  resource: {2}
  timestamp: "2021-03-02T23:37:16.390Z"
  severity: "NOTICE"
  logName: "projects/amigs-minecraft/logs/cloudaudit.googleapis.com%2Factivity"
  receiveTimestamp: "2021-03-02T23:37:17.604618532Z"
}
```

Showing logs for last 1 hour starting at 3/2/21, 5:37 PM.

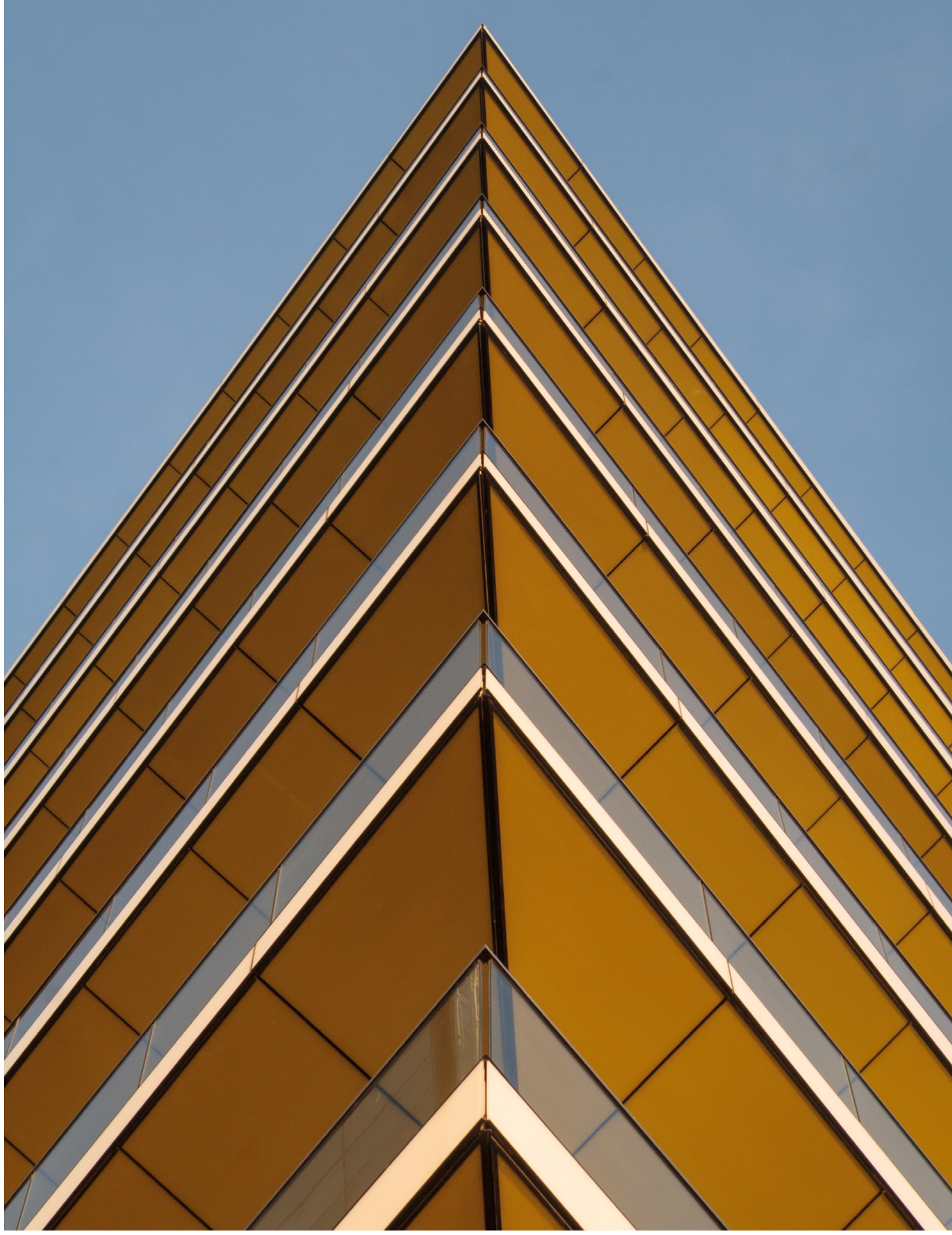
Scope by project — Search based on the logs from this project only. Scoping by project allows you to query only the logs that are generated by this project regardless of where the logs are stored.

Scope by storage [✓] — Search based on one or more logs storage views. Scoping by views allows you to query logs buckets within this project or logs routed from this project to buckets in other projects.

- SECOPS-PROD / _DEFAULT
  - _AllLogs — projects/secops-prod/locations/global/buckets/_Default/views/_AllLogs
  - _Default — projects/secops-prod/locations/global/buckets/_Default/views/_Default
- SECOPS-PROD / _REQUIRED
  - _AllLogs — projects/secops-prod/locations/global/buckets/_Required/views/_AllLogs
- [✓] SECOPS-PROD / ALL-AUDIT-LOGS-BUCKET
  - [✓] _AllLogs — projects/secops-prod/locations/global/buckets/all-audit-logs-bucket/views/...

Audit logs should now be stored in your central project but not the child projects. You might wonder about the audit logs going to the `_Required` sink. We can't change the audit logs going to the `_Required` sink (see section three of our companion paper: [Data governance: Principles for securing and managing logs]() for more details) but these logs are typically low volume and there is no charge for storing them in the `_Required` bucket.

Google Cloud



03

Compliance challenge 3:
Configuring retention for a log bucket

Customer example:

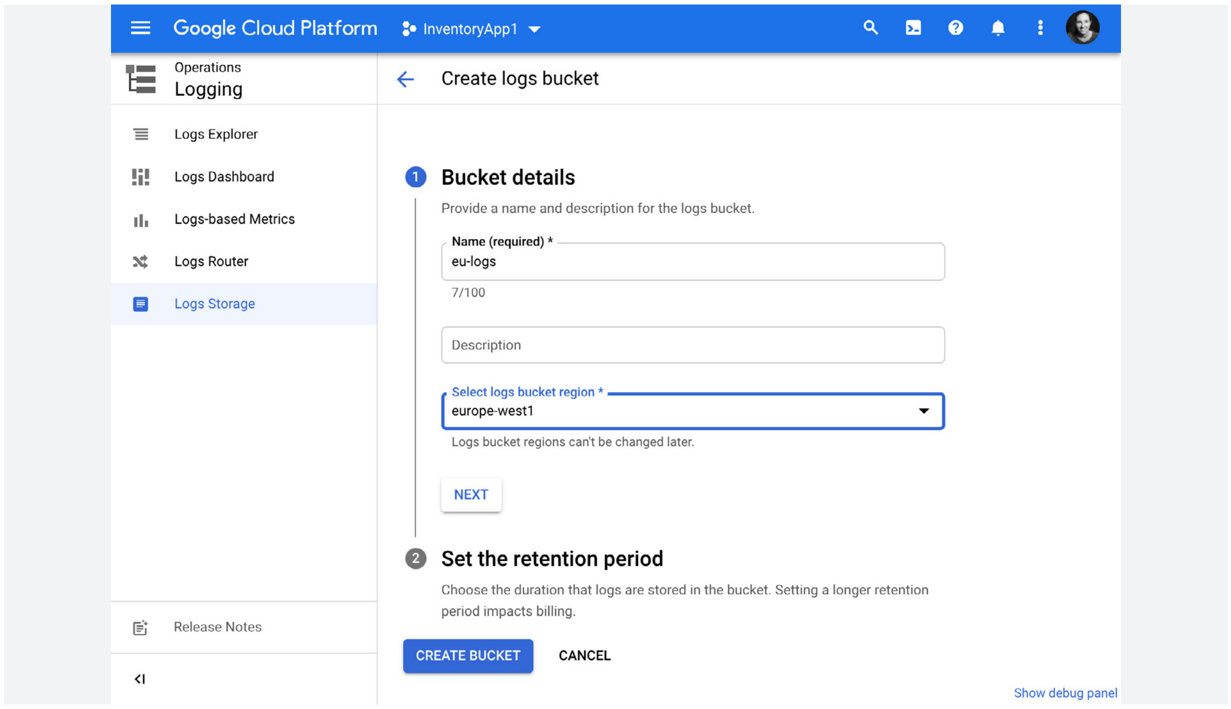
A retailer in the US was handling credit card transactions and they needed to retain logs for 13 months for PCI compliance.

Solution:

Unlike other log management systems which require shifting data between hot and cold storage, Cloud Logging easily scales to handle months or years of data using intelligent partitioning by time, so setting up retention is a single step.

01

Edit the retention on a log bucket — From the Logs Storage page, choose edit on a bucket. Set the retention period to 400 days or whatever number of days you prefer as shown below. Note that you won't be able to edit the `_Required` bucket so if you need to retain these logs for longer periods of time, you'll want to create a copy in a different bucket.



Create logs bucket

1 Bucket details

Provide a name and description for the logs bucket.

Name (required) *: eu-logs

7/100

Description

Select logs bucket region *: europe-west1

Logs bucket regions can't be changed later.

NEXT

2 Set the retention period

Choose the duration that logs are stored in the bucket. Setting a longer retention period impacts billing.

CREATE BUCKET CANCEL

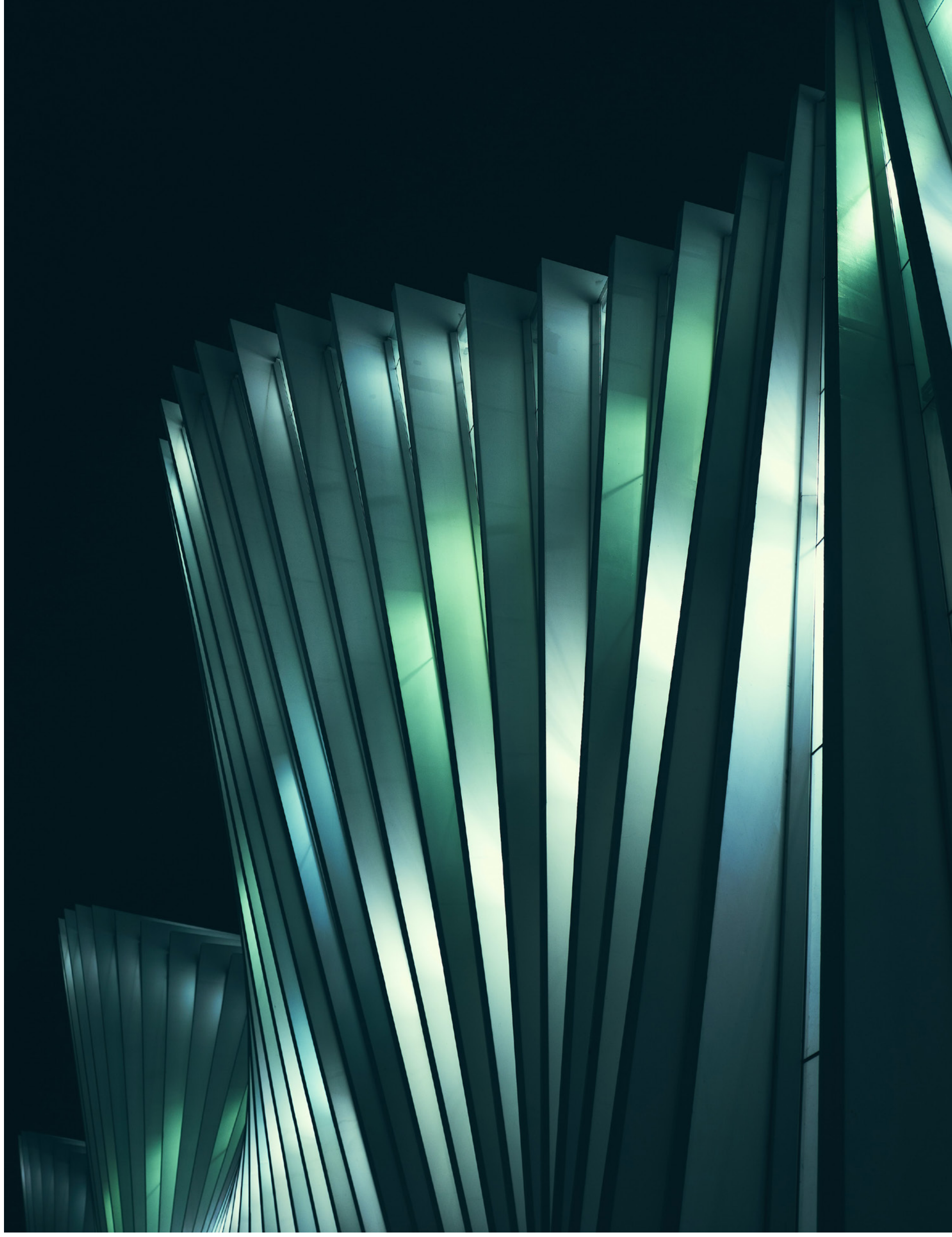
02

OPTIONAL - protect the logs from any tampering by locking the bucket — If you need to prove to auditors that the bucket was not tampered with, you may choose to lock a log bucket.

WARNING — Locking a log bucket is ***irreversible*** and the bucket cannot be deleted nor can logs in a bucket be deleted until the retention policy has expired *even if sensitive data is accidentally logged to this bucket*. Therefore, **locking a bucket should be used only where clear requirements exist and when you can confirm that the data flowing into this bucket will not need to be deleted or modified for other compliance requirements.**

```
$ gcloud logging buckets update all-audit-logs-bucket
--location=global --locked
```

Google Cloud



04

Compliance challenge 4: Managing access to logs

Customer example:

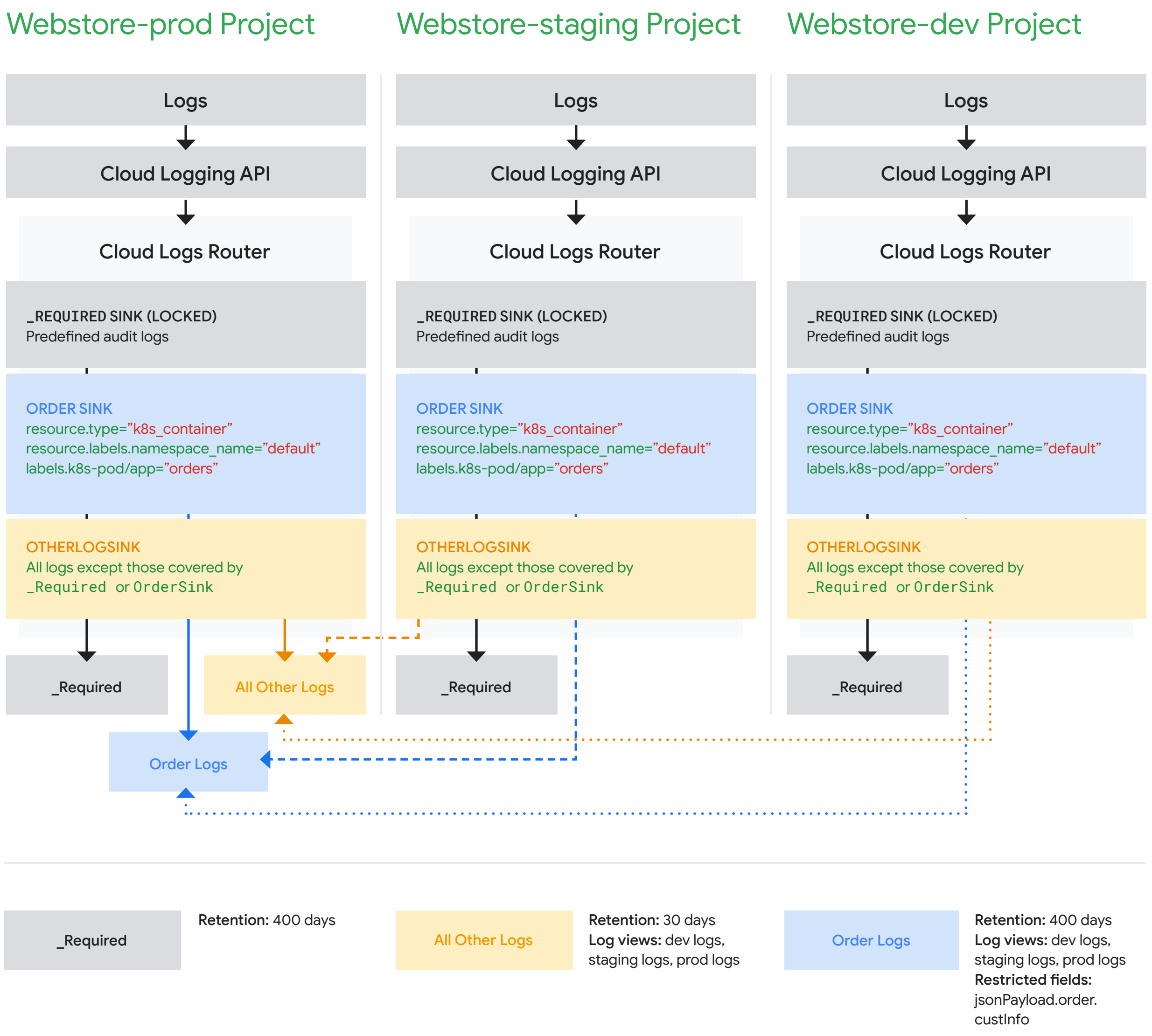
The final example will build on all the previous solutions. A Canadian retailer running a webstore with three projects for prod, staging and dev, needed to grant different teams access to different logs. They also needed to restrict access to user PII (email, address, phone, etc.) unless needed for troubleshooting based on their internal policies. All logs had to be kept in the Canadian region, `northamerica-northeast1`.

Solution:

Separate data based on a mix of log buckets, log views and restricted fields.

01

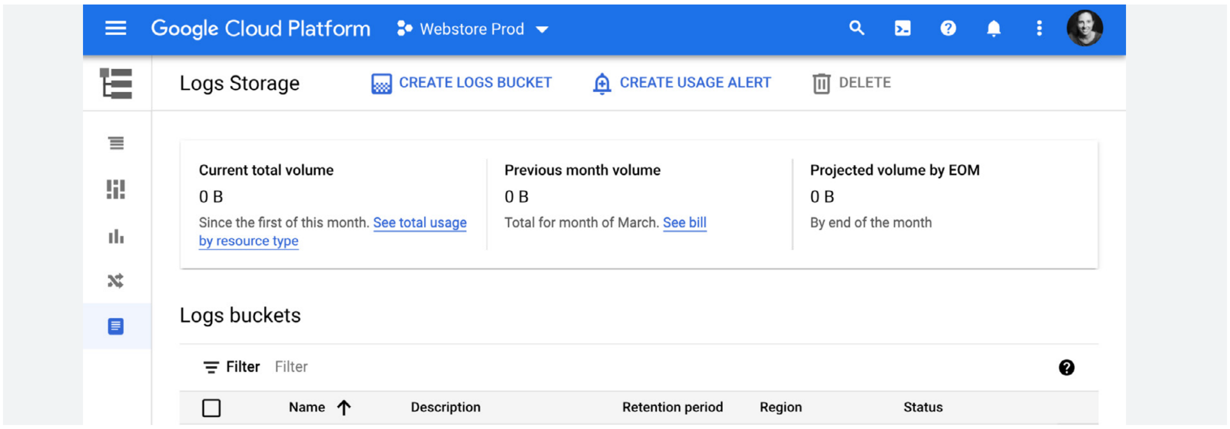
Your first step is to spend a bit of time planning your desired end state. You need to centralize logs from several projects into a single pane of glass for debugging, manage retention, control what regions logs are stored in, and manage access to logs. Based on those goals, see the example architecture diagram:


Webstore-prod Project
Webstore-staging Project
Webstore-dev Project
Logs
Cloud Logging API
Cloud Logs Router
_REQUIRED SINK (LOCKED)
Predefined audit logs
ORDER SINK
resource.type="k8s_container"
resource.labels.namespace_name="default"
labels.k8s-pod/app="orders"
OTHERLOGSINK
All logs except those covered by _Required or OrderSink
_Required
All Other Logs
Order Logs
Retention: 400 days
Retention: 30 days
Log views: dev logs, staging logs, prod logs
Retention: 400 days
Log views: dev logs, staging logs, prod logs
Restricted fields: jsonPayload.order.custInfo

In the example architecture you have three projects: `Webstore-prod`, `Webstore-staging` and `Webstore-dev`. The advantage of splitting logs into multiple buckets is that it gives you more flexibility around dividing the logs vs. just using log views, so you can subdivide the logs only along certain predefined dimensions.

02

Now it is time to build the desired end state you mapped out above. Create the two new log buckets in your `Webstore-prod` project, named `OrdersLogs` and `AllOtherLogs`, in the region of your choice (`northamerica-northeast1` in the customer example). To do this, go to the Logs Storage section of the Cloud Console and click on “Create logs bucket”, name the bucket and choose the correct region. Repeat for the second bucket as well.



Current total volume	Previous month volume	Projected volume by EOM
0 B	0 B	0 B
Since the first of this month. See total usage by resource type	Total for month of March. See bill	By end of the month

Logs buckets

Filter Filter

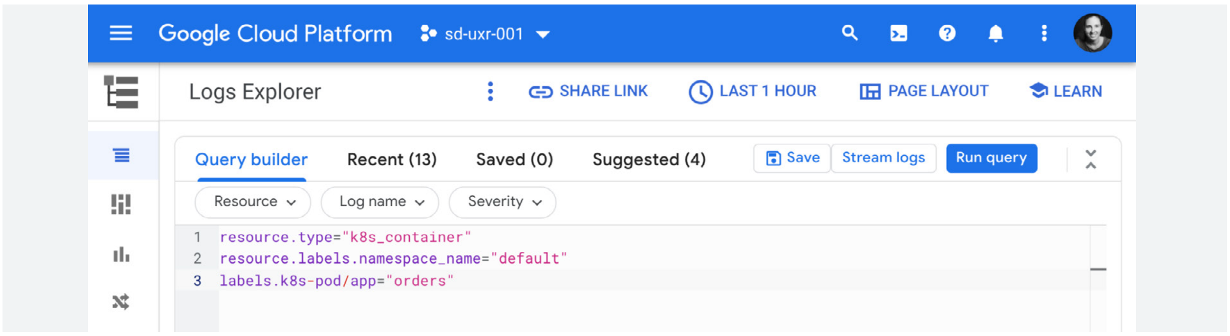
Name	Description	Retention period	Region	Status

03

Use log sinks to send the correct logs to the correct bucket. In this example, separate orders logs based on the GKE workload name:

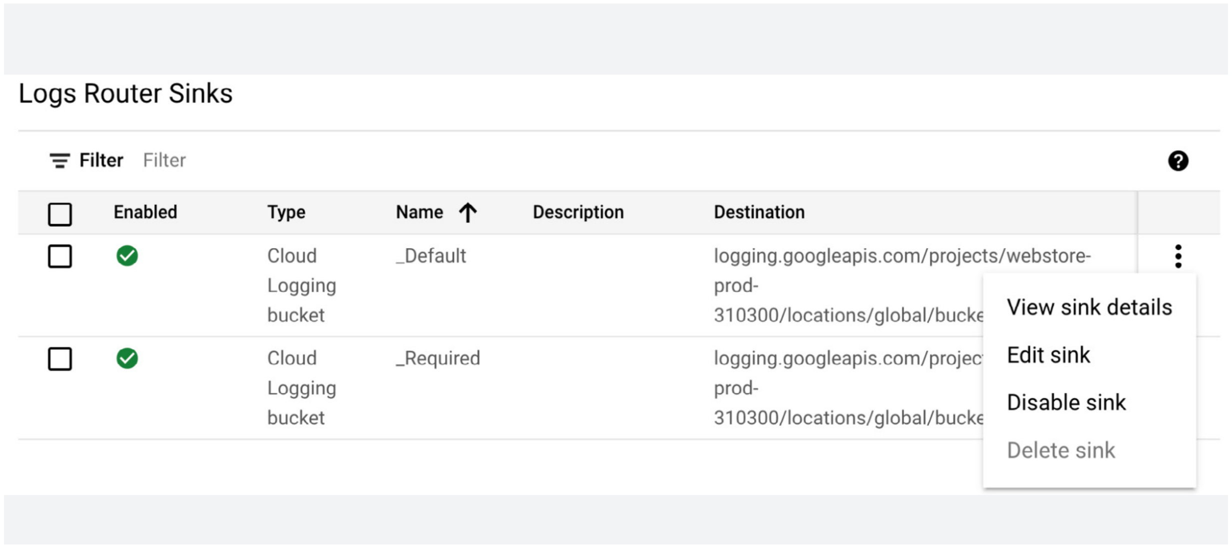
```
resource.type=”k8s_container”
resource.labels.namespace_name=”default”
labels.k8s-pod/app=”orders”
```

- Check in Logs Explorer to make sure you’re capturing the correct logs:



```
resource.type="k8s_container"
resource.labels.namespace_name="default"
labels.k8s-pod/app="orders"
```

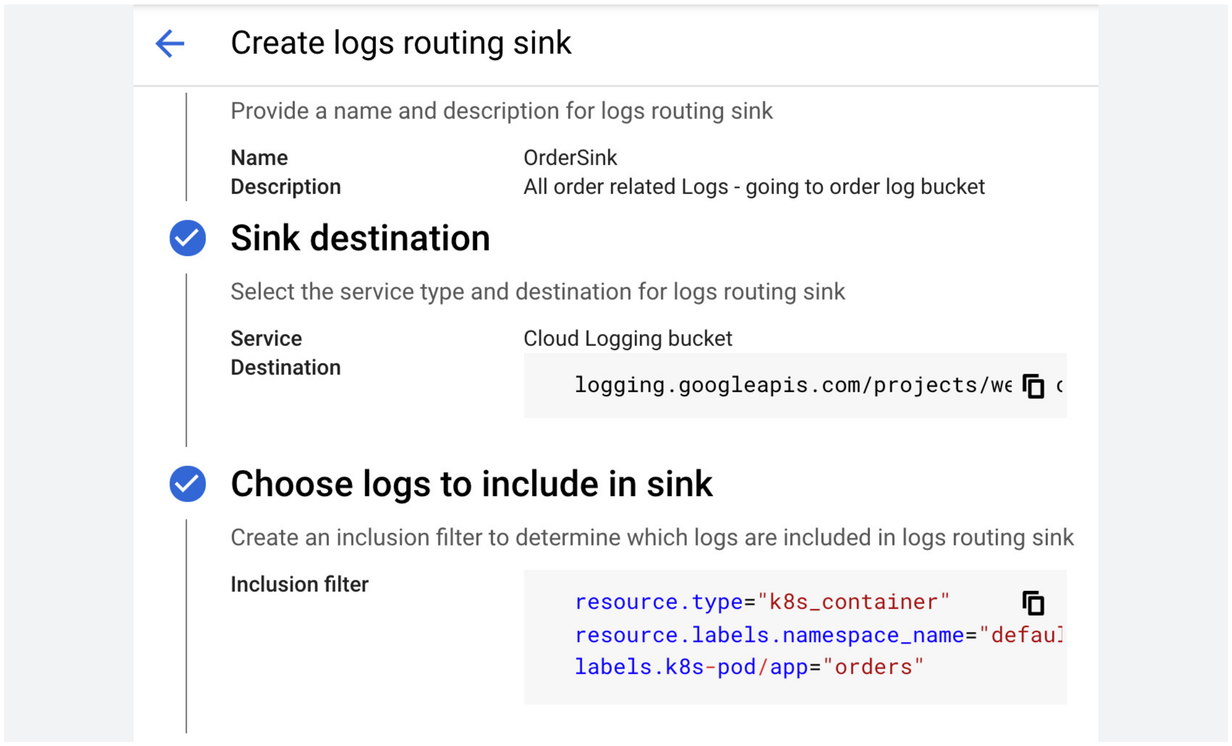
- Now that you have identified the correct filter for the logs you want, disable the `_Default` sink and replace it with two other sinks, `OrderSink` and `OtherLogSink`, one for each of your buckets. Disable the `_Default` sink in the Logs Router.



Logs Router Sinks

	Enabled	Type	Name ↑	Description	Destination
☐	✓	Cloud Logging bucket	_Default		logging.googleapis.com/projects/webstore-prod-310300/locations/global/bucke
☐	✓	Cloud Logging bucket	_Required		logging.googleapis.com/projec prod-310300/locations/global/bucke

- Create `OrderSink` first by selecting Create Sink in the Log Router. Under Sink destination, select Cloud Logging bucket and choose the newly created `OrdersLogs` as the destination. Paste the filter for the order logs into the inclusion filter.



Create logs routing sink

Provide a name and description for logs routing sink

Name	OrderSink
Description	All order related Logs - going to order log bucket

Sink destination

Select the service type and destination for logs routing sink

Service	Cloud Logging bucket
Destination	logging.googleapis.com/projects/we

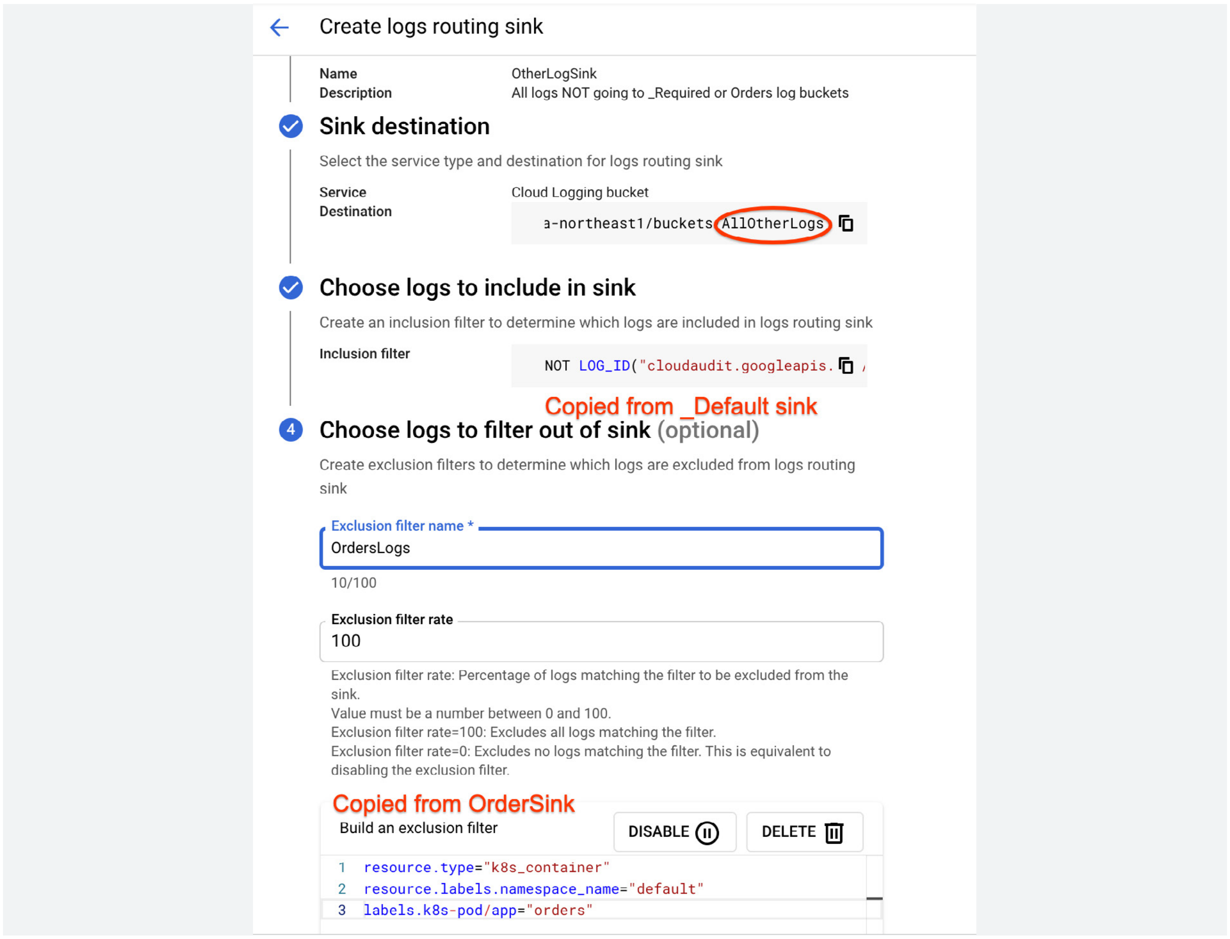
Choose logs to include in sink

Create an inclusion filter to determine which logs are included in logs routing sink

Inclusion filter

```
resource.type="k8s_container"
resource.labels.namespace_name="defaul
labels.k8s-pod/app="orders"
```

- Repeat the process of creating a new sink with **`OtherLogSink`**. Under Sink destination, select Cloud Logging bucket and choose the newly created **`AllOtherLogs`** as the destination. This sink should capture all logs that aren't going to the **`_Required`** bucket or the **`OrdersLogs`** bucket. You can do this by building an inclusion filter from the **`_Default`** sink and an exclusion filter from the filter you used with **`OrderSink`** as shown below:



Create logs routing sink

Name: OtherLogSink
Description: All logs NOT going to _Required or Orders log buckets

Sink destination

Select the service type and destination for logs routing sink

Service: Cloud Logging bucket
Destination: a-northeast1/buckets/AllOtherLogs

Choose logs to include in sink

Create an inclusion filter to determine which logs are included in logs routing sink

Inclusion filter: `NOT LOG_ID("cloudaudit.googleapis.`

Copied from _Default sink

Choose logs to filter out of sink (optional)

Create exclusion filters to determine which logs are excluded from logs routing sink

Exclusion filter name *: OrdersLogs
10/100

Exclusion filter rate: 100

Exclusion filter rate: Percentage of logs matching the filter to be excluded from the sink.
Value must be a number between 0 and 100.
Exclusion filter rate=100: Excludes all logs matching the filter.
Exclusion filter rate=0: Excludes no logs matching the filter. This is equivalent to disabling the exclusion filter.

Copied from OrderSink

Build an exclusion filter

```
resource.type="k8s_container"
resource.labels.namespace_name="default"
labels.k8s-pod/app="orders"
```

- Repeat the process of disabling the **`_Default`** sink and adding two new sinks to your other two projects, **`Webstore-staging`** and **`Webstore-dev`**. Be sure to choose "Use a logs bucket in another project" and fill in the appropriate project, location and bucket:

  logging.googleapis.com/projects/**webstore-prod**/locations/**northamerica-northeast1**/buckets/**OrdersLogs**

  and

  logging.googleapis.com/projects/**webstore-prod**/locations/**northamerica-northeast1**/buckets/**AllOtherLogs**

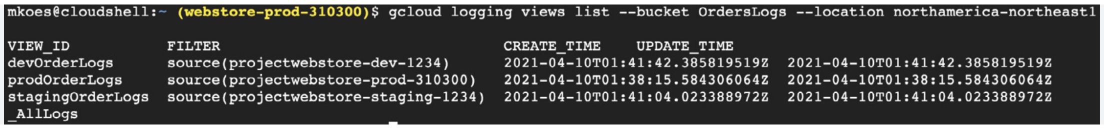
04

At this point, logs from all three projects are flowing into your two new log buckets in the Webstore-Prod project. Now, create log views to divide the logs by environment in the **OrdersLogs** and **AllOtherLogs** log buckets. Since there is not yet support for creating log views in the Cloud Console, create the log views via gcloud. You'll need to use the ID for each project:

```
$ gcloud logging views create prodOrderLogs --bucket OrdersLogs
--location northamerica-northeast1
--log-filter="source(projectwebstore-prod-1234)"
--description="Webstore-Prod Order Logs"
```

Once you're done, you can view all the log views with the following command:

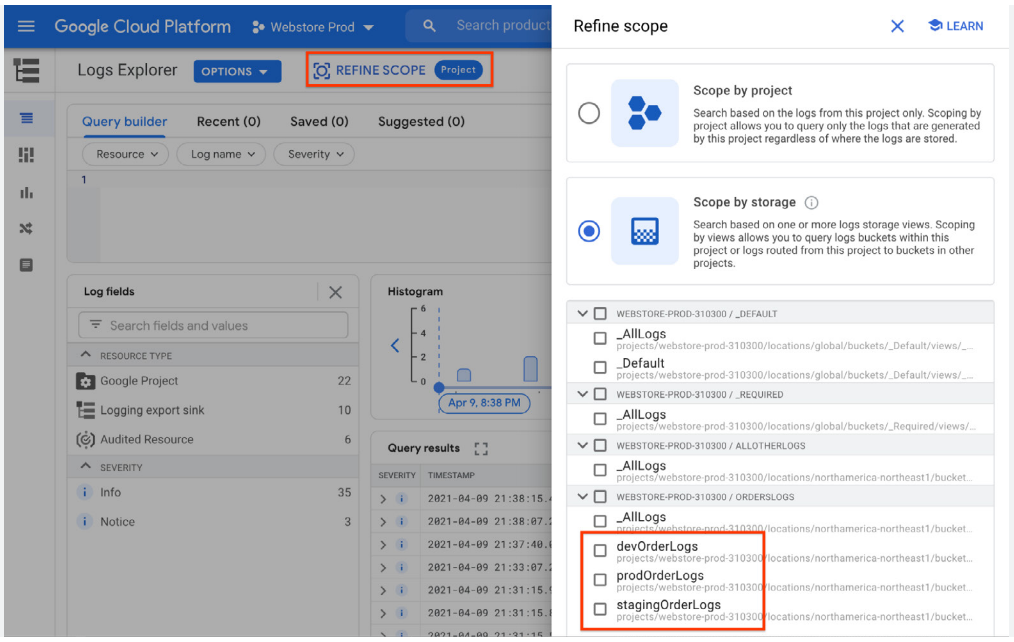
```
$ gcloud logging views list --bucket OrdersLogs
--location northamerica-northeast1
```



```
mkoes@cloudshell:~ (webstore-prod-310300)$ gcloud logging views list --bucket OrdersLogs --location northamerica-northeast1

VIEW_ID           FILTER                                  CREATE_TIME                  UPDATE_TIME
devOrderLogs      source(projectwebstore-dev-1234)        2021-04-10T01:41:42.385819519Z  2021-04-10T01:41:42.385819519Z
prodOrderLogs     source(projectwebstore-prod-310300)     2021-04-10T01:38:15.584306064Z  2021-04-10T01:38:15.584306064Z
stagingOrderLogs  source(projectwebstore-staging-1234)    2021-04-10T01:41:04.023388972Z  2021-04-10T01:41:04.023388972Z
_AllLogs
```

You should also see the new log views in the Log Explorer:



Refine scope

( ) Scope by project
Search based on the logs from this project only. Scoping by project allows you to query only the logs that are generated by this project regardless of where the logs are stored.

(●) Scope by storage
Search based on one or more logs storage views. Scoping by views allows you to query logs buckets within this project or logs routed from this project to buckets in other projects.

WEBSTORE-PROD-310300 / _DEFAULT
- _AllLogs projects/webstore-prod-310300/locations/global/buckets/_Default/views/_...
- _Default projects/webstore-prod-310300/locations/global/buckets/_Default/views/_...

WEBSTORE-PROD-310300 / _REQUIRED
- _AllLogs projects/webstore-prod-310300/locations/global/buckets/_Required/views/...

WEBSTORE-PROD-310300 / ALLOTHERLOGS
- _AllLogs projects/webstore-prod-310300/locations/northamerica-northeast1/bucket...

WEBSTORE-PROD-310300 / ORDERSLOGS
- _AllLogs projects/webstore-prod-310300/locations/northamerica-northeast1/bucket...
- devOrderLogs projects/webstore-prod-310300/locations/northamerica-northeast1/bucket...
- prodOrderLogs projects/webstore-prod-310300/locations/northamerica-northeast1/bucket...
- stagingOrderLogs projects/webstore-prod-310300/locations/northamerica-northeast1/bucket...

```
2021-04-09 21:38:15.
2021-04-09 21:38:07.
2021-04-09 21:37:40.
2021-04-09 21:33:07.
2021-04-09 21:31:15.
2021-04-09 21:31:15.
```

Repeat for the **AllOtherLogs** bucket.

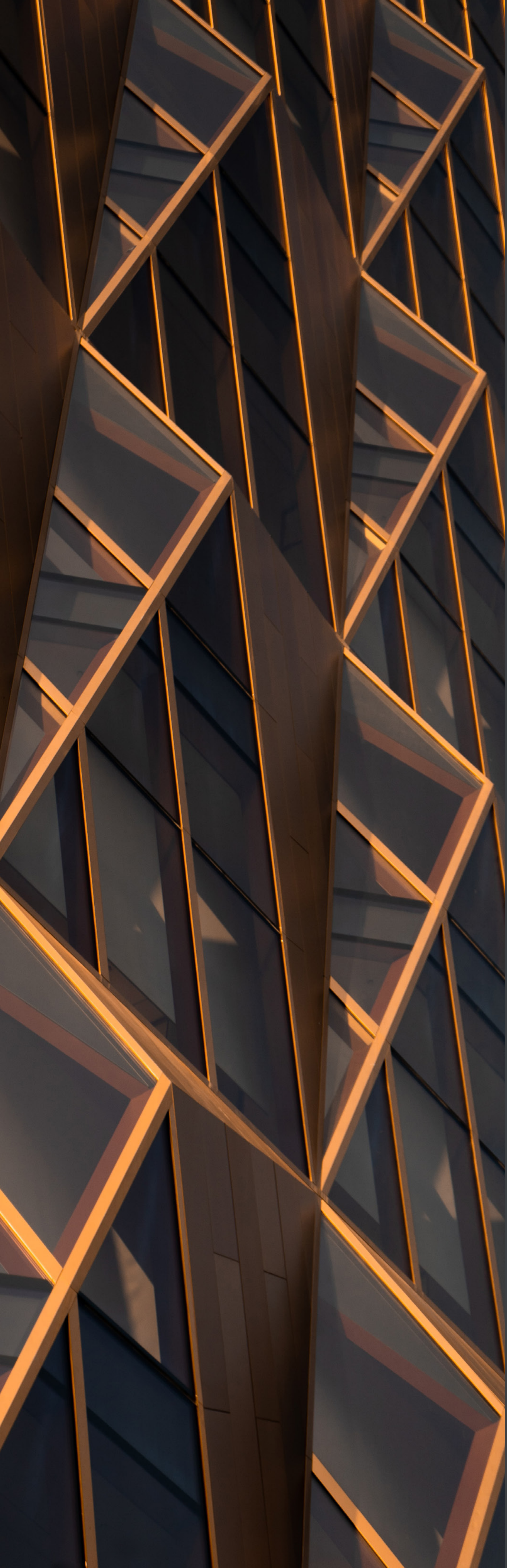
05

Finally, restrict access to the sensitive fields in the `OrdersLogs` bucket. Since there is not yet support for managing sensitive fields in the Cloud Console, use gcloud. This feature is still in Preview as of May 2021:

```
$ gcloud alpha logging buckets update OrdersLogs
--location northamerica-northeast1
--restricted-fields “jsonPayload.order.custInfo”
```

06

You’re now ready to add users to the new log views and grant permissions for the sensitive fields using standard Google Cloud IAM permissions based on the data they need for their roles.



Conclusion

Logs provide valuable data for troubleshooting, metrics, governance and security. Logs are subject to data governance regulations and guidelines and this paper provided step by step instructions for how to solve four of the most common challenges. If you need more information about Cloud Logging with regard to data governance, or would like to view best practices from Google's product team on how to configure your logs collection system, download our companion whitepaper Data governance: Principles for securing and managing logs.

If you would like to provide feedback or have questions about this guide or other Cloud Logging topics, please join the discussion group.

Appendix

At a Glance: Best practices for compliance across the logging lifecycle

Although different customers have different compliance needs, we've collected 14 best practices that are a great start for most customers in addressing their compliance requirements.

Lifecycle stage	Best practice
Log generation & collection	Encourage developers to use structured logs where possible. This allows you to more easily analyze your logs and promotes better logging hygiene, reducing the chance of logging sensitive information such as passwords. It also makes it possible to restrict access to sensitive data in your logs.
	Use severity levels consistently within your application so that you can easily identify actionable log messages. This also makes it easier to control costs by downsampling less important logs.
	Within Google Cloud, most platform logs and audit logs are collected automatically, simplifying the compliance effort. A few high volume logs are off by default and do need to be manually enabled. Turn on data access audit logs across your Google Cloud organization and VPC flow logs for critical subnetworks.
	Install a logging agent on all VMs, whether on GCP, other clouds, or on prem and configure it to collect key logs.

Lifecycle stage	Best practice
Log processing & routing	When sending logs from outside of Google Cloud, use a regional endpoint, e.g. `europe-west2-logging.googleapis.com`, to guarantee that a log entry is received in a specific region.
	Use an aggregated log sink at the org or folder level to collect logs from all projects, especially for security use cases. It's especially important to centralize logs that are important for compliance (e.g. audit logs, PCI logs, etc.) in a logging project so that even if the projects generating the logs are deleted, the logs will be available.
	Automate the setup of log sinks and log buckets for new projects to comply with your requirements around where logs are stored.
Log storage and retention	Create a custom log bucket for the region in which you need to store your logs and redirect your log sinks to the new log bucket instead of the global `_Default` bucket. Automate this step for new projects. The `_Required` bucket only contains logs about system events, Google actions and administrative actions (Create, Update, Delete) on GCP APIs.
	Set up organization policies so that all new custom log buckets will be created in approved regions.
	Configure retention on a log bucket to match your compliance requirements.
	If you have compliance requirements to prove logs have not been modified or deleted, consider locking the log bucket. However, keep in mind that this cannot be undone should requirements change in the future or if unintended data is sent to this log bucket so this is best used with well structured logs that you control.
Log analysis and access	Grant users only access to logs they need for their job.
	Use log views to subdivide access within a log bucket if your users can divide access to logs by source project, resource type, or log name. For other delineations, create different log buckets.
	Use field level access control to deny access to sensitive fields by default.