

# GCP and CSPM-GCP Integration

This Google Cloud integration collects and analyzes a wide range of logs and metrics to provide comprehensive visibility into your cloud environment. It ingests **Firewall Logs**, **VPC Flow Logs**, **DNS Logs**, and **Load Balancing Logs** exported from **Cloud Logging** via a **Pub/Sub topic sink**. Additionally, it gathers detailed **metrics and metadata** from **Google Cloud Monitoring** across core services, including **Compute Engine**, **Cloud SQL**, **Cloud Run**, **GKE**, **Firestore**, **Dataproc**, **Pub/Sub**, **Redis**, **Storage**, **Load Balancing**, and **Billing**. This enables in-depth monitoring of infrastructure, application performance, network activity, and cost trends.

## Logs

- **Firewall Logs:** Record allowed and denied network traffic based on firewall rules.
- **VPC Flow Logs:** Capture IP traffic flowing to and from network interfaces in a VPC.
- **DNS Logs:** Track DNS queries and responses handled by Google Cloud DNS.
- **Load Balancing Logs:** Provide request-level logs of traffic handled by load balancers, including latency and backend info.

---

## Metrics

- **GCP Billing Metrics:** Track resource usage and cost across GCP services.
- **GCP Compute Metrics:** Monitor performance of Compute Engine instances (CPU, memory, disk, etc.).
- **GCP Firestore Metrics:** Provide insights into Firestore usage like reads, writes, and storage.
- **GCP Load Balancing Metrics:** Measure load balancer traffic, request counts, latency, and backend health.
- **GCP Storage Metrics:** Report usage, operation counts, and latency for Cloud Storage buckets.
- **GCP GKE Metrics:** Monitor Kubernetes clusters including node health, pod usage, and resource consumption.
- **GCP Dataproc Metrics:** Track job status, cluster usage, and Hadoop/Spark performance in Dataproc.
- **GCP PubSub Metrics:** Show message throughput, subscription rates, and processing latency.
- **GCP Redis Metrics:** Display memory usage, operations per second, and cache hit/miss rates for Memorystore Redis.
- **GCP Cloud Run Metrics:** Measure request counts, container instance metrics, and response times.

- **GCP CloudSQL Metrics:** Provide visibility into database performance, including connections, query latency, and CPU usage.

## Authentication

To use this Google Cloud Platform (GCP) integration, you need to set up a **Service Account** with a **Role** and a **Service Account Key** to access data on your GCP project.

### 1. Service Account

First, you need to [create a Service Account](#). A Service Account (SA) is a particular type of Google account intended to represent a non-human user who needs to access the GCP resources.

The AQUILA Agent uses the SA to access data on Google Cloud Platform using the Google APIs.

### 2. Required IAM Service Account Roles:

#### For CSPM-GCP Integration

- **Browser:** This role grants read access to the project hierarchy.
- **Cloud Asset Viewer:** Can view asset metadata across GCP services.

#### For GCP Integration

- **Cloud Memorystore Redis Viewer:** Can view configuration and metadata of Redis instances.
- **Cloud SQL Viewer:** Can view Cloud SQL instance metadata and settings, but not data.
- **Compute Viewer:** Can view all Compute Engine resources (instances, disks, etc.) but not modify them.
- **Logs Viewer:** Can view logs in Cloud Logging across the project.
- **Monitoring Viewer:** Can view monitoring dashboards, alerts, and metrics in Cloud Monitoring.
- **Private Logs Viewer:** Can view all logs, including those with restricted data (e.g., data access logs).
- **Pub/Sub Subscriber:** Grants permission to receive and acknowledge messages from Pub/Sub subscriptions.
- **Viewer:** Read-only access to all resources in a project.

### 3. Logs Collection Configuration

With a properly configured Service Account and the integration setting in place, it's time to start collecting some logs.

#### Requirements

You need to create a few dedicated Google Cloud resources before starting, in detail:

- **Pub/Sub Topic:** A messaging endpoint where publishers send messages that can then be delivered to one or more subscribers.
- **Subscription:** A configuration attached to a Pub/Sub topic that delivers messages to subscribers, either by push or pull.
- **Log Sink:** A configuration that routes logs from Cloud Logging to a specified destination such as Pub/Sub, Cloud Storage, or BigQuery.

It's recommended to have a separate Pub/Sub topics for each of the log types so that they can be parsed and stored in a specific data stream.

Here's an example of collecting Audit Logs using a Pub/Sub topic, a subscription, and a Log Router. We will create the resources in the Google Cloud Console and then configure the Google Cloud Platform integration.

## Example Setup Using Google Cloud Console

1. Navigate to "**Logging**" > "**Log Router**" > "**Create Sink**".
2. Provide a **Sink name** and description.
3. For **Sink destination**, select "**Cloud Pub/Sub topic**". Choose an existing topic or create a new one.
4. If a new topic is created, you must also **create a subscription** for it.
5. Under "**Choose logs to include in sink**", use a filter like:  
logName:"cloudaudit.googleapis.com"

## 4. Enable API Services

- **Cloud Asset API:** Provides metadata inventory and history of GCP resources and IAM policies for security analysis, audit, and compliance.
- **Cloud SQL Admin API:** Enables programmatic management of Cloud SQL instances, including creation, configuration, and backups.
- **Memorystore for Redis API:** Allows automated management of Redis instances on Memorystore, including provisioning, scaling, and configuration.

## 5. Service Account Key

Next, with the Service Account (SA) with access to Google Cloud Platform (GCP) resources setup, you need some credentials to associate with it: a Service Account Key.

From the list of SA (Service Accounts):

1. Go to **IAM & Admin** > **Service Accounts** in the GCP Console.
2. Click the service account you created.
3. Under the "**Keys**" section, click "**Add Key**" > "**Create new key**".
4. Choose **JSON** as the key type.
5. **Download and securely store** the generated private key (it cannot be retrieved again from GCP if lost).

## Please provide the following information to CyTech:

- **Project ID** - The Project ID is the Google Cloud project ID where your resources exist.
- **Credentials File** - Save the JSON file with the private key in a secure location of the file system, and make sure that the Log Collector Agent has at least read-only privileges to this file. Specify the file path in the Log Collector Agent integration UI in the "Credentials File" field. For example: /home/ubuntu/credentials.json.
- **Pub/Sub Topic** - Name of the topic where the logs are written to.
- **Subscription** - Use the short subscription name here, not the full-blown path with the project ID. You can find it as "Subscription ID" on the Google Cloud Console.

---

Revision #5

Created 9 May 2025 05:24:27 by Richmond Abella

Updated 9 October 2025 01:57:23