Google: Cloud asset inventory

Cloud Asset Inventory provides inventory services based on a time series database. Cloud Asset Inventory allows you to:

- Search asset metadata by using a custom query language
- Export all asset metadata at a certain timestamp or export event change history during a specific timeframe
- Monitor asset changes by subscribing to real-time notifications
- Analyze IAM policy to find out who has access to what

The Splunk GCP Application Template is a blueprint of visualizations, reports, and searches focused on Google Cloud use cases. Many of the reports within GCP require Google Cloud asset inventory data to be periodically generated and sent into Splunk. You'll need to create an inventory generation pipeline so you can populate GCP's dashboards and reports.

The GCP Application Template only leverages data generated by the batch export data API, part of the batch export view of asset data provided by Google.

There are three solutions you can use to ingest Google Cloud asset inventory data into Splunk, but choosing which solution is best depends on your specific needs.

Solutions

Option 1 - Pull from bucket

You can use Cloud Scheduler to trigger a Cloud Function on a regular schedule. This Cloud Function then sends a batch export request to the Asset Inventory API. This results in a bulk export of the current cloud asset inventory to a Cloud Storage bucket. Finally, the Splunk Add-on for Google Cloud Platform (GCP-TA) is configured to periodically monitor and ingest new files appearing in the Cloud Storage bucket.

The components you'll need to use for this solution are:

- Cloud Scheduler
- Cloud Pub/Sub
- Cloud Function (asset export)
- Cloud Storage

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
• **GCP-TA storage input**

This solution avoids Splunk HEC whitelisting considerations, and comes with the advantage that GCP-TA is developed and supported by Splunk. It also takes minimal Google Cloud infrastructure cost to implement. However, be aware that large inventories may exceed file size limitations of GCP-TA GCS input (~250 MB).

If you're just getting started and want to get the reports up and running, this solution could be a good fit for you. The number of moving pieces is minimal and it is relatively easy to set up. It’s also inexpensive in comparison to other approaches and will likely take you pretty far.

You can view [detailed setup instructions](#) for this solution in the GitHub repository.

### Option 2 - Dataflow

You can leverage Dataflow batch and streaming jobs to facilitate the delivery of asset inventory data to a Splunk HEC. This approach uses Cloud Scheduler to regularly trigger a Cloud Function, which in turn is responsible for initiating an Asset Inventory API bulk export to Cloud Storage. Another Cloud Function receives an event trigger when the export operation is complete. This function then starts a batch Dataflow job which converts newline-delimited JSON files into Pub/Sub messages and publishes them to a topic. In parallel, a streaming Dataflow pipeline is also running, which subscribes to the topic and delivers them to a Splunk HEC.

The components you’ll need to use for this solution are:

• Cloud Scheduler
• Cloud Function (asset export)
• Asset Inventory API
• Cloud Storage
• Cloud Function (launch Dataflow batch job)
• [Batch Dataflow job](#)
• Pub/Sub
• [Streaming Dataflow job](#)
• Splunk HEC

This solution uses dataflow templates developed and supported by Google. It’s a natural choice for environments already leveraging streaming Dataflow to Splunk HEC log delivery pipeline Dead letter topic support for undeliverable messages. Delivery backlog is easy to monitor through Google Cloud Pub/Sub metrics, and HEC tokens are also encrypted using KMS. There are some downsides to this solution, however. As well as the setup for this solution being more complex than the other solutions described previously, be aware that both Dataflow templates are in beta, pre-GA launch stage. There also are additional costs of running a Dataflow cluster that you’ll need to account for.
If you're already using Dataflow to stream Cloud Logging events into Splunk, then this solution could be a good fit for you. If you want to scale out your infrastructure and need to make sure what you deploy is easily monitorable, horizontally-scalable, and fault-tolerant, then this solution will allow you to take advantage of that same infrastructure investment to deliver asset inventory data to Splunk.

You can view detailed setup instructions for this solution in the GitHub repository.

**Option 3 - Serverless push-to-Splunk**

You can use Cloud Functions not only for triggering an export of asset inventory data, but also to perform the delivery to a Splunk HEC. Cloud Scheduler regularly triggers a Cloud Function, which in turn is responsible for initiating an Asset Inventory API bulk export to Cloud Storage. A second Cloud Function is configured to trigger on bucket object create/finalize events. This function splits the exported files into smaller files if necessary, and deliver them directly to a Splunk HEC. If the delivery fails, the messages are placed into a Pub/Sub topic for later redelivery attempts.

The components you’ll need to use for this solution are:

- Cloud Scheduler
- **Cloud Function (asset export)**
- Asset Inventory API
- Cloud Storage
- **Cloud Function (Cloud Storage ingest)**
- Pub/Sub
- **Cloud Function (delivery retry)**
- Splunk HEC

This solution should incur minimal Google Cloud infrastructure costs, and it allows for automated redelivery attempts of deadletters. The GCP Application Template is primarily tested with this solution as it is written by the same author, however be aware that this is **not a Splunk or Google supported solution**.

If solution 2 is not within your reach or is too much for your needs, this solution could be a good fit for you. It has some of the same fault-tolerant and horizontally scalable properties that Dataflow shares, but without the cost of running a batch streaming Dataflow job. However, because neither Google or Splunk support can assist in the operation of this solution, be aware that you could find yourself "on your own" should things go wrong. This means that for production pipelines, this solution likely isn’t appropriate - however for other environments, it could be a better fit.

You can view detailed setup instructions for this solution in the GitHub repository.
Additional resources

The content in this guide comes from a previously published blog, one of the thousands of Splunk resources available to help users succeed. In addition, these Splunk resources might help you understand and implement this use case:

- App: GCP Application Template for Splunk
- .Conf Talk: Harnessing the power of Splunk and Google Cloud: deploy, ingest, and beyond
- Splunk Docs: Connect to GCP

Application

When your Splunk deployment is ingesting Google Cloud asset inventory data, you can use the data to achieve a number of security and observability use cases