>

Assets are an organization's Google Cloud resources, like Compute Engine instances or Cloud Storage buckets. This guide shows you how to use Security Command Center client libraries to access metadata about an organization's assets.

Before you set up a source, you'll need to complete the following:

- Set up a service account and SDK (/security-command-center/docs/how-to-programmatic-access)

All Security Command Center list APIs are paginated. Each response returns a page of results and a token to return the next page. The page size is configurable. The default pageSize is 10, it can be set to a minimum of 1, and maximum of 1000.

These examples show how to list all assets for an organization:

An organization might have a lot of Assets. The example above doesn't use any filters, so all assets are returned. Security Command Center enables you to use asset filters to get information about specific assets. Filters are like "where" clauses in SQL statements, except instead of columns, they apply to the objects returned by the API.

The example below uses a filter to return only project resources in the organization:

The previous examples show how to list an organization's current set of assets. Security Command Center also enables you to view a historical snapshot of an organization's assets. The example below

returns the state of all resources at a specific point in time. Security Command Center supports millisecond time resolutions.

Security Command Center enables you to compare an asset to a previous period in time to identify if it was created, deleted, or present during the specified time period. This is captured in the `state_change` field on the list asset results.

Following are some other useful asset filters.

These example filters match assets created at or before July 18th, 2019 at 8:26:21PM GMT. With the `create_time` filter, you can express time using the following formats and types:

- Unix time (in milliseconds) as an integer literal

- RFC 3339 as a string literal

`$USER` is typically in the format `user:someone@domain.com`. The comparison for `user` uses the substring operator `:` and an exact match isn't necessary.

`$PROJECT_1_NAME` and `$PROJECT_2_NAME` are resource identifiers in the form of `//cloudresourcemanager.googleapis.com/projects/$PROJECT_ID`, where `$PROJECT_ID` is the project number. A complete example would be something like: `//cloudresourcemanager.googleapis.com/projects/100090906`

This filter returns Compute Engine images that contain substring "Debia":