Mandiant

ASM GOOGLE CLOUD INTEGRATION

This document outlines the Google Cloud integration process with Mandiant Advantage Attack Surface Management (MA-ASM). Using this integration, you gain visibility of your Google Cloud assets and extended ecosystem for faster identification of new or unmanaged assets and remediation of vulnerabilities, misconfigurations, and exposures.

For customers with IT infrastructure across multiple clouds, Mandiant recommends using the inbound integrations with AWS, Azure, and Google Cloud for a consolidated view of cloud and internet-facing assets.

The MA-ASM Google Cloud integration leverages Google Cloud's service account impersonation. By creating a service account and delegating your account access, you'll be able to assume control of the service account and fetch the respective resources. This prevents both parties from being required to retain their credentials.

- MA-ASM implements a form of **confused-deputy (https://en.wikipedia.org/wiki/Confused_deputy_problem)** mitigation, which works by preventing the creation of the integration if an integration associated with the same service account already exists.
- If your Google Cloud organization is configured with an organization policy that restricts identities by domain, such as `iam.allowedPolicyMemberDomains`, you need to allow MA-ASM's Google Workspace Customer ID in the policy. This Customer ID is *C0439tmya.* For instructions on how to allow a specific Customer ID, see **Setting the organization policy (https://cloud.google.com/resource-manager/docs/organization-policy/restricting-domains#setting_the_organization_policy)**.

See **Google Cloud Related Entities in MA-ASM** to learn more about the Entities created in MA-ASM using this integration.

Create a Service Account

Follow either Method A or Method B to create a Google Cloud service account.

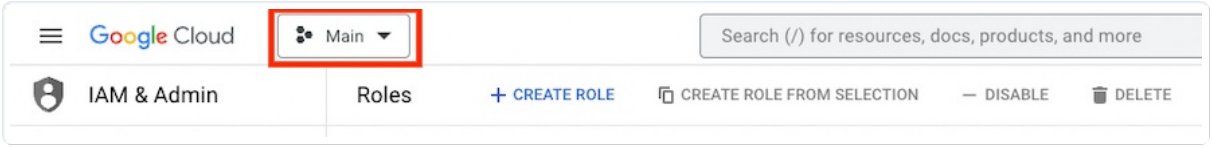
- **Method A: Create a Service Account through Google Cloud Console**
- **Method B: Create a Service Account through gcloud CLI**

Once a service account has been created, proceed to the instructions to **Create Google Cloud Integration within MA-ASM**.

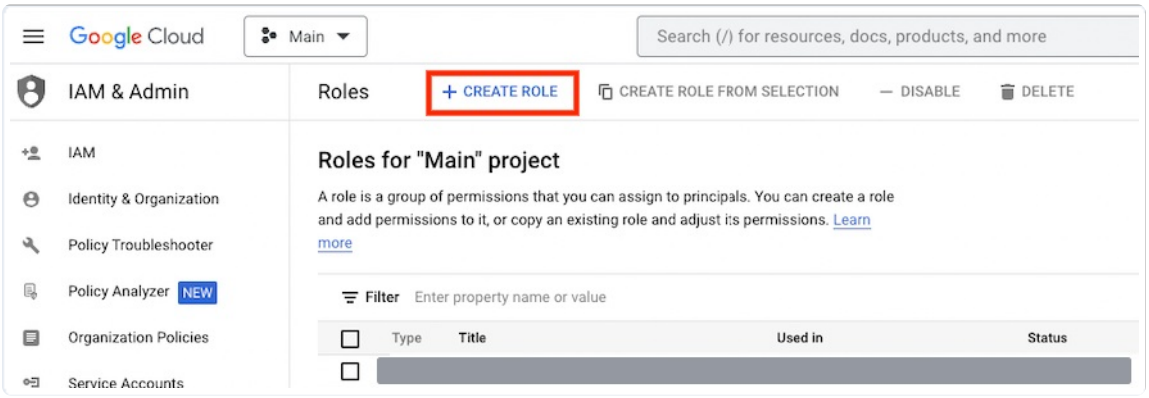
Method A: Create a Service Account through Google Cloud Console

Execute the following steps in the Google Cloud's web console:

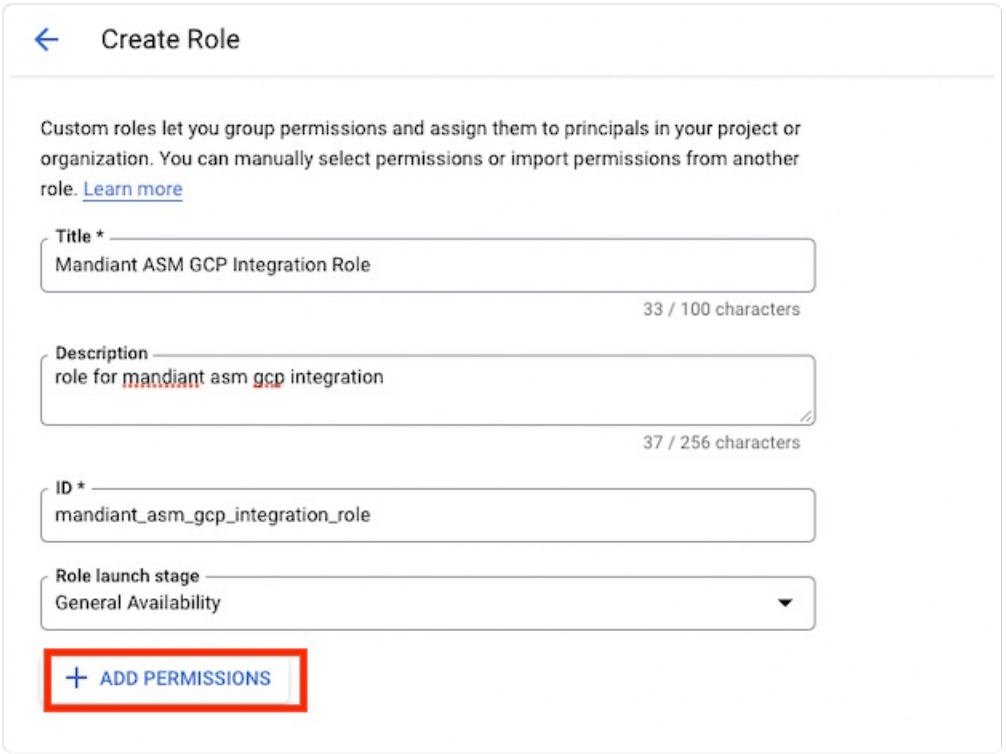
1. Once successfully authenticated with Google Cloud console through a user who has the appropriate permissions to create a service account and a role, browse to **https://console.cloud.google.com/iam-admin/roles**. Ensure the current project is the one you would like for the integration to access. If you are not in the correct project, navigate to the desired project.



2. Create a Role using the following process:
    a. Click **+ Create role**.

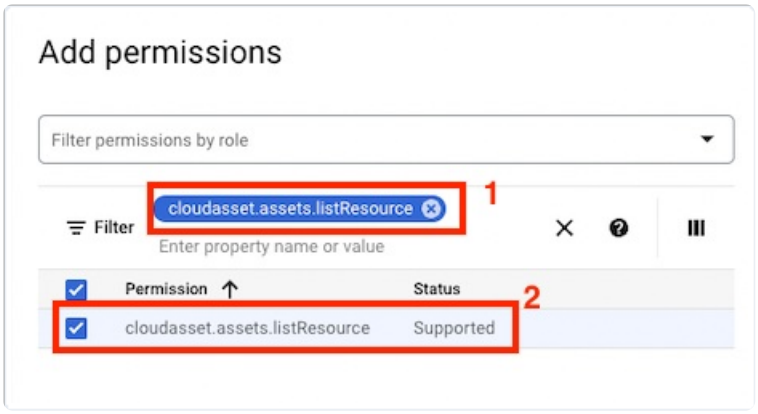


    b. In the **Create Role** interface, fill in the following fields and click **+ Add permissions**.
        - **Title**: Friendly title for the role, for example, *Mandiant ASM GCP Integration Role*
        - **Description**: Short description describing what the role does
        - **ID**: ID for the role, for example, *mandiant_asm_gcp_integration-role*
        - **Role launch stage**: General Availability

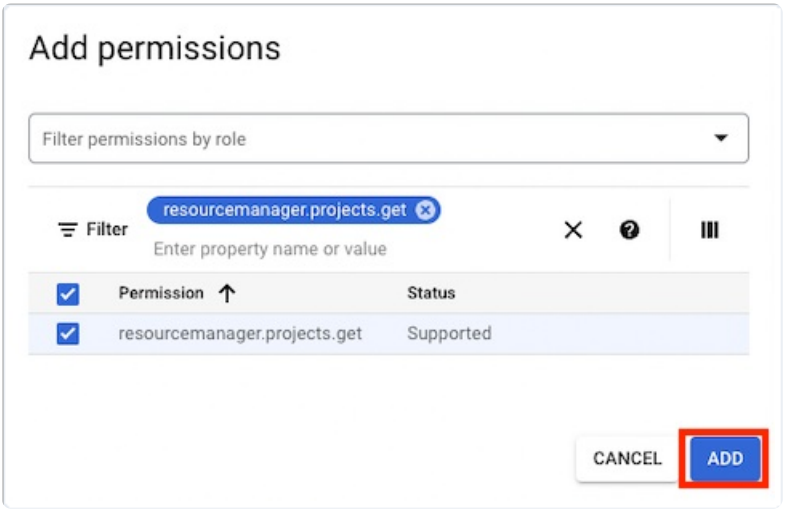


    c. In the **Add permissions** sub-menu, add the following permissions:
        - `cloudasset.assets.listResource`
        - `dns.managedZones.list`
        - `dns.resourceRecordSets.list`
        - `apigateway.apiconfigs.get`
        - `resourcemanager.projects.get`

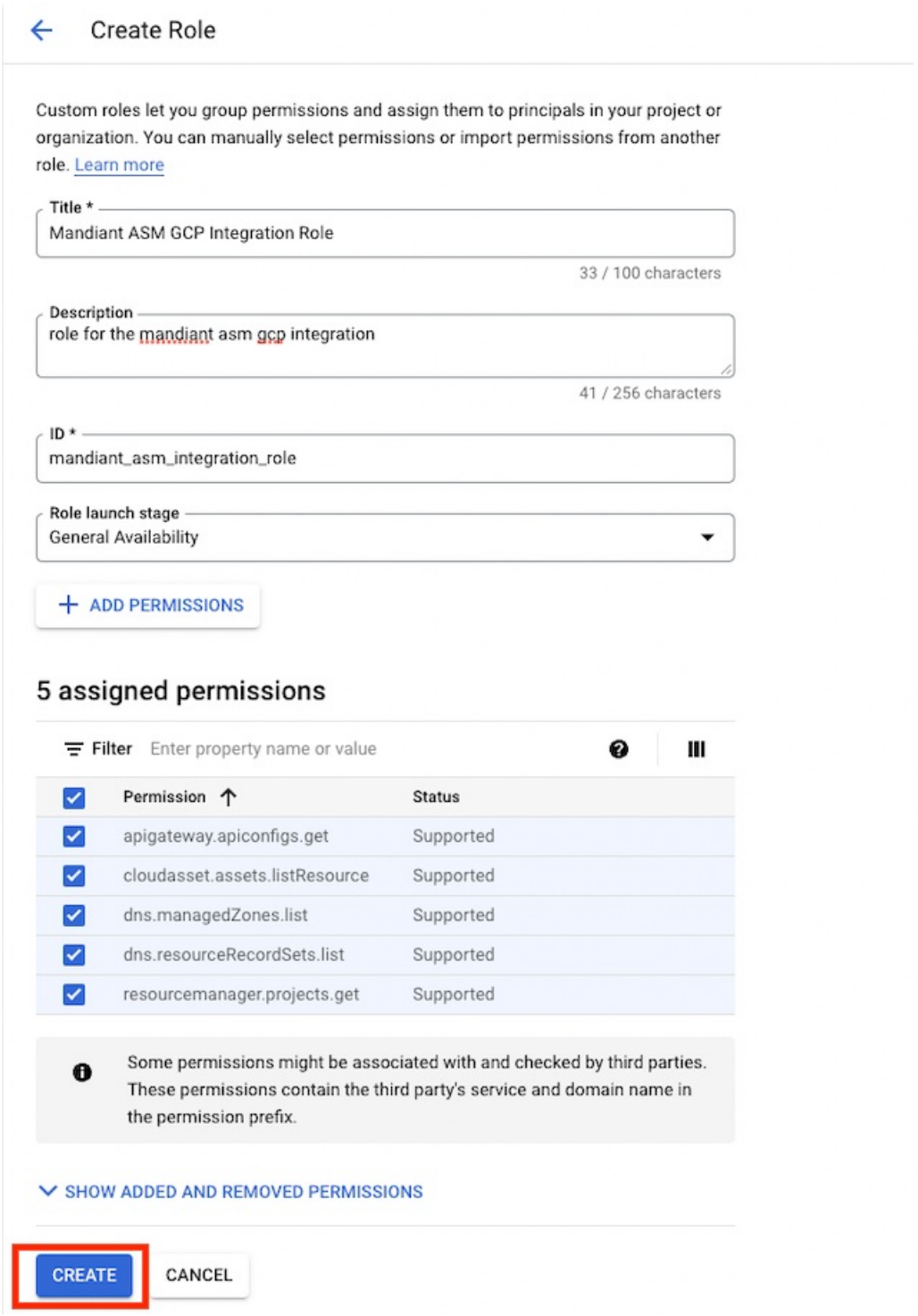
- The easiest way to do this is to copy the permission name, for example, `cloudasset.assets.listResource` and paste it into the **Enter property name or value** section and then click `enter`. When the role appears, click the checkbox next to it. Repeat this process for each permission.
- After clicking the checkbox, clear the filter before searching for the next permission, or else the new permission won't show up.



d. After all the permissions have been added, click **Add**.

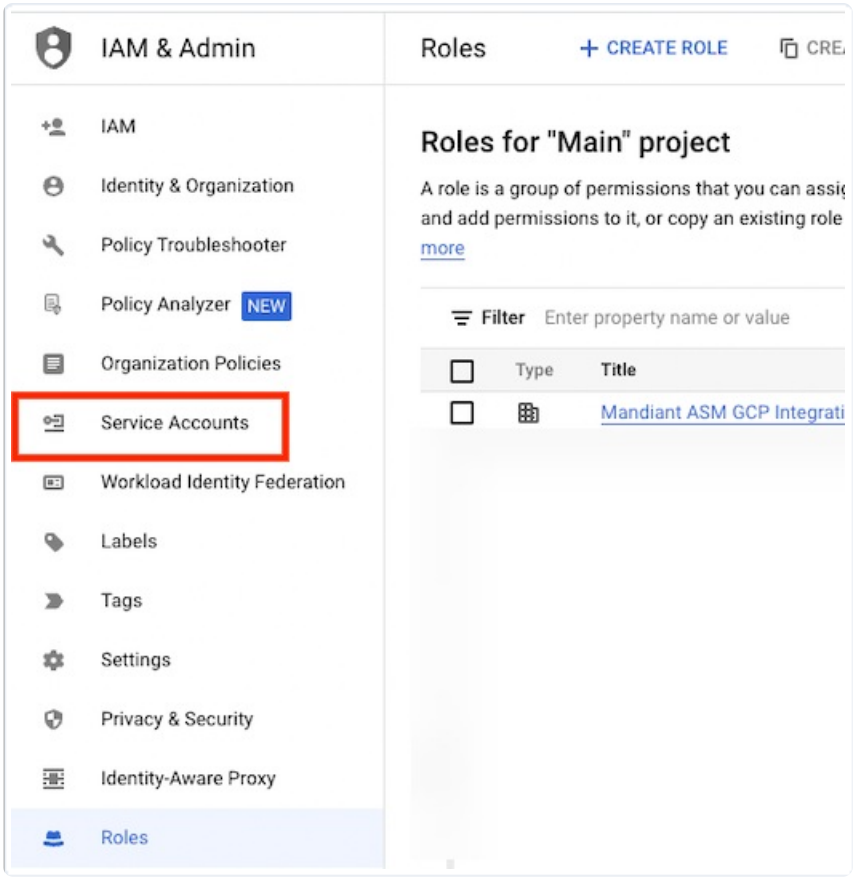


e. Confirm the **Create Role** interface shows that all five permissions have been assigned and click **Create**.

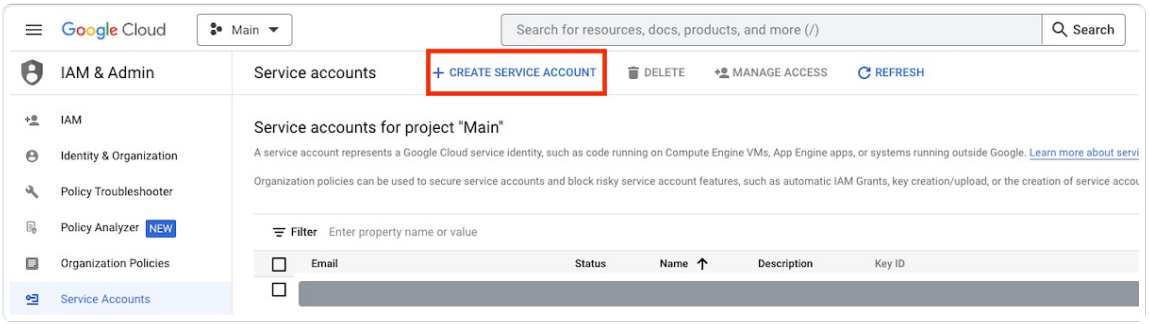


Create Role interface showing role details including five assigned permissions.

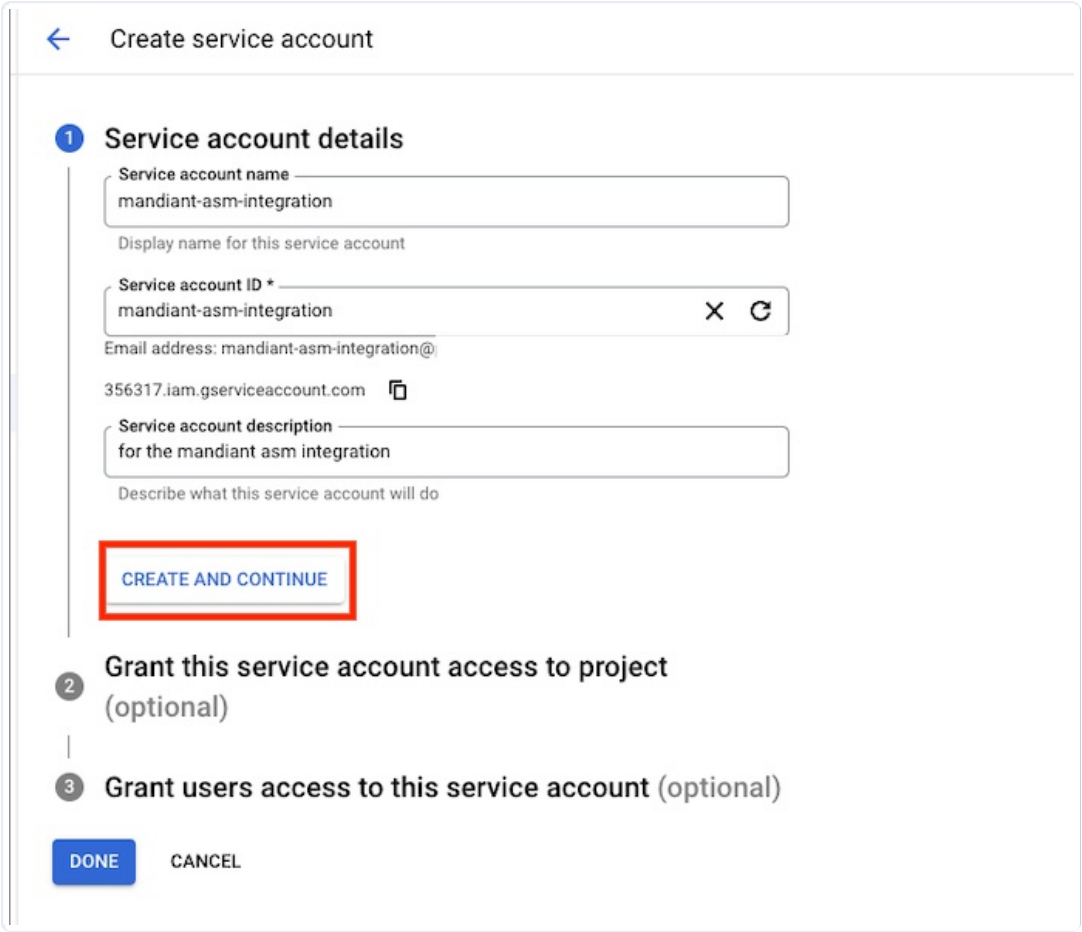
3. Create a service account using the following process:
    a. From the **IAM & Admin** menu, click the **Service Accounts** tab.



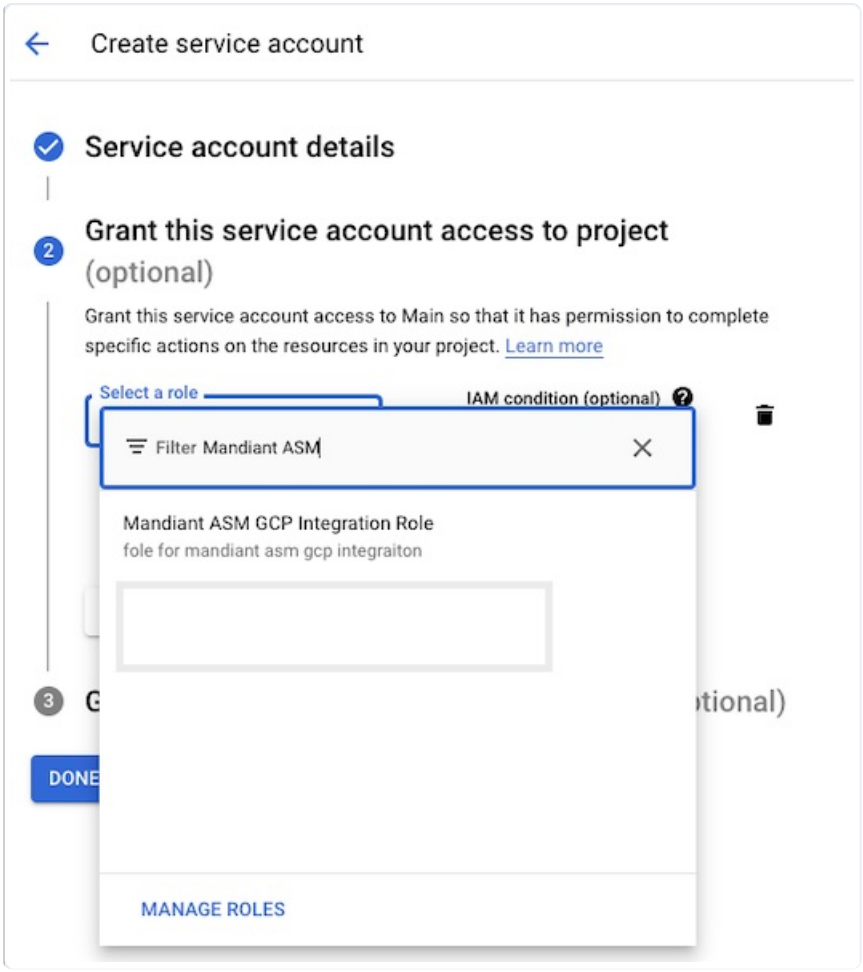
b. Click **+ Create service account**.



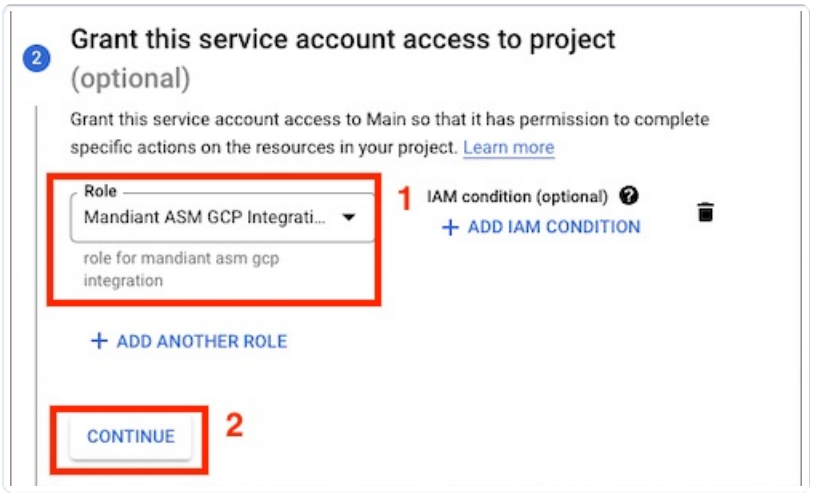
c. In the **Create service account** interface, fill in the following fields for the **Service account details** section and click **Create and continue**.
   - **Service account name**: Friendly name for the service account.
   - **Service account ID**: This is automatically populated based on the first field.
   - **Service account description**: Quick note providing a description of what this service account does.



d. For the **Grant this service account access to project** section, click **Select a role** and, in the filter, type the name of the role created earlier in this section, for example, *Mandiant ASM GCP Integration Role*.

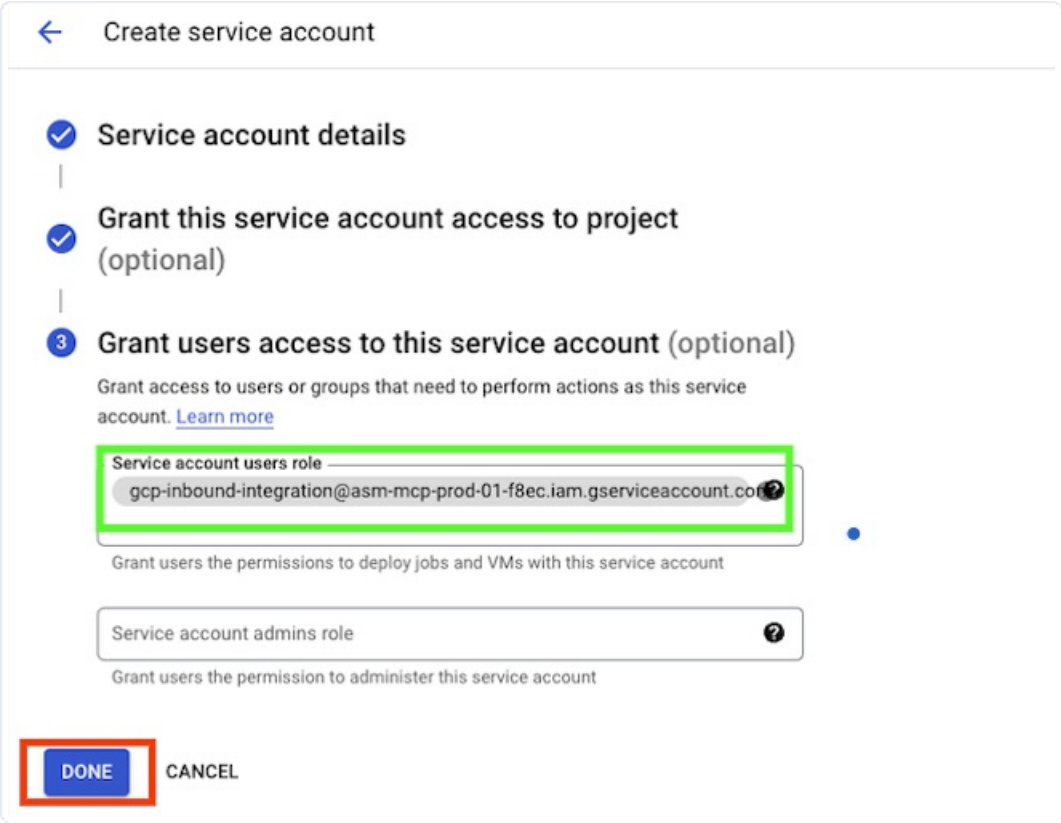


e. Select the role and click **Continue**.

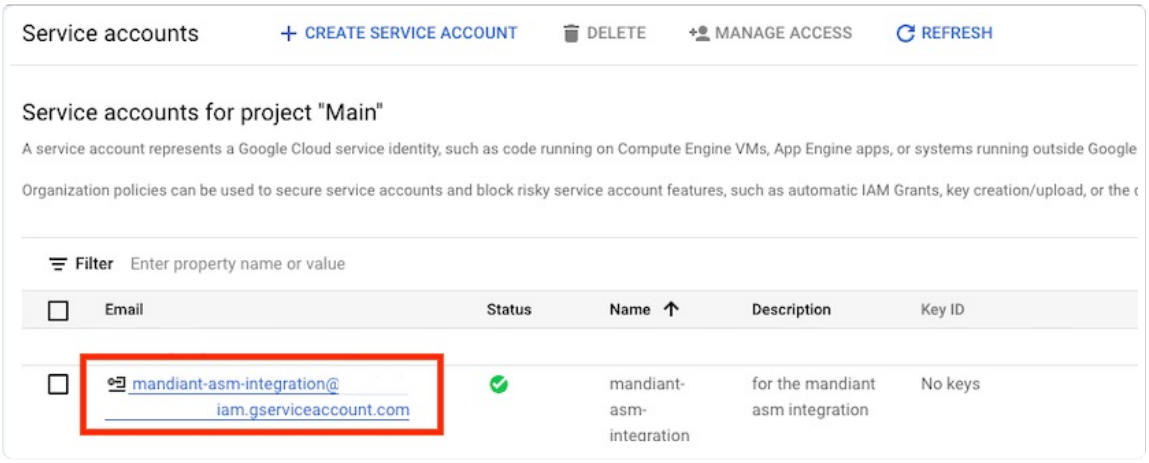


f. Under the **Grant users access to this service account** section, for the **Service account users role** field, enter the Mandiant Google Cloud service account email and click **Done**.

> The Mandiant Google Cloud service account email is `gcp-inbound-integration@asm-mcp-prod-01-f8ec.iam.gserviceaccount.com` .

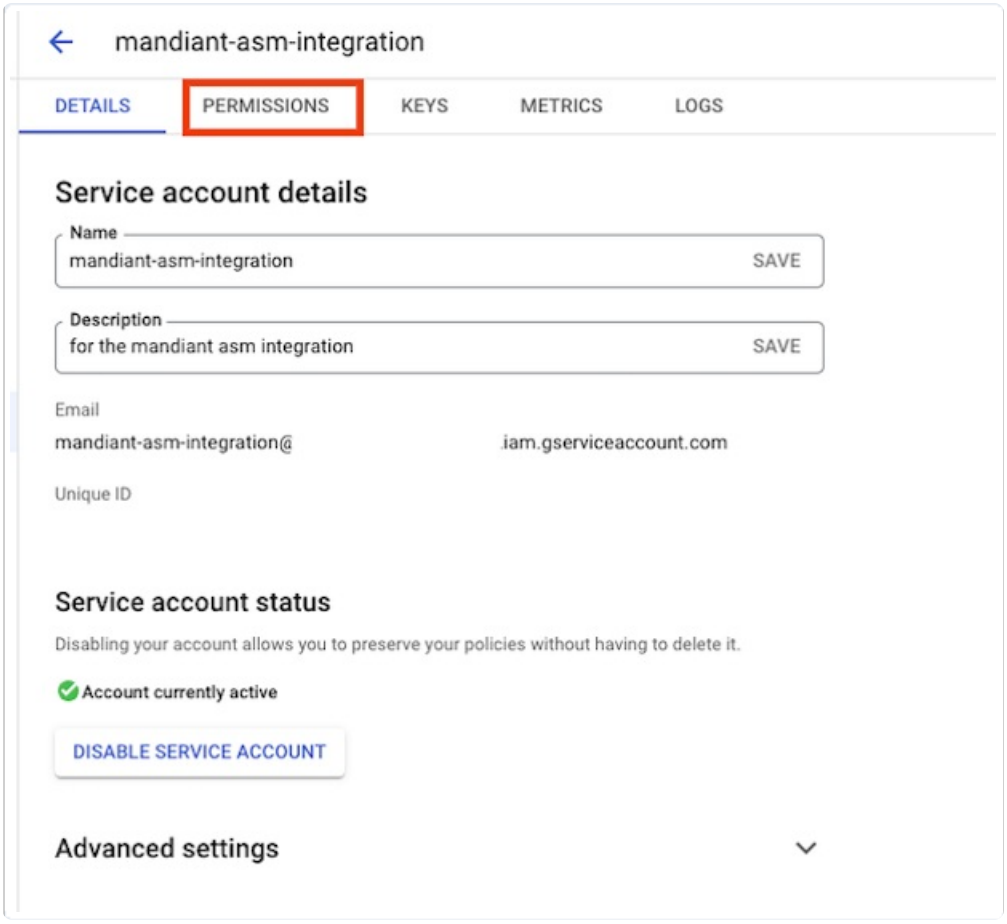


g. Once the Service Account is created, navigate to the **Service accounts** page (you are automatically redirected after clicking **Done** in the previous step), and click the email link belonging to the service account you just created.

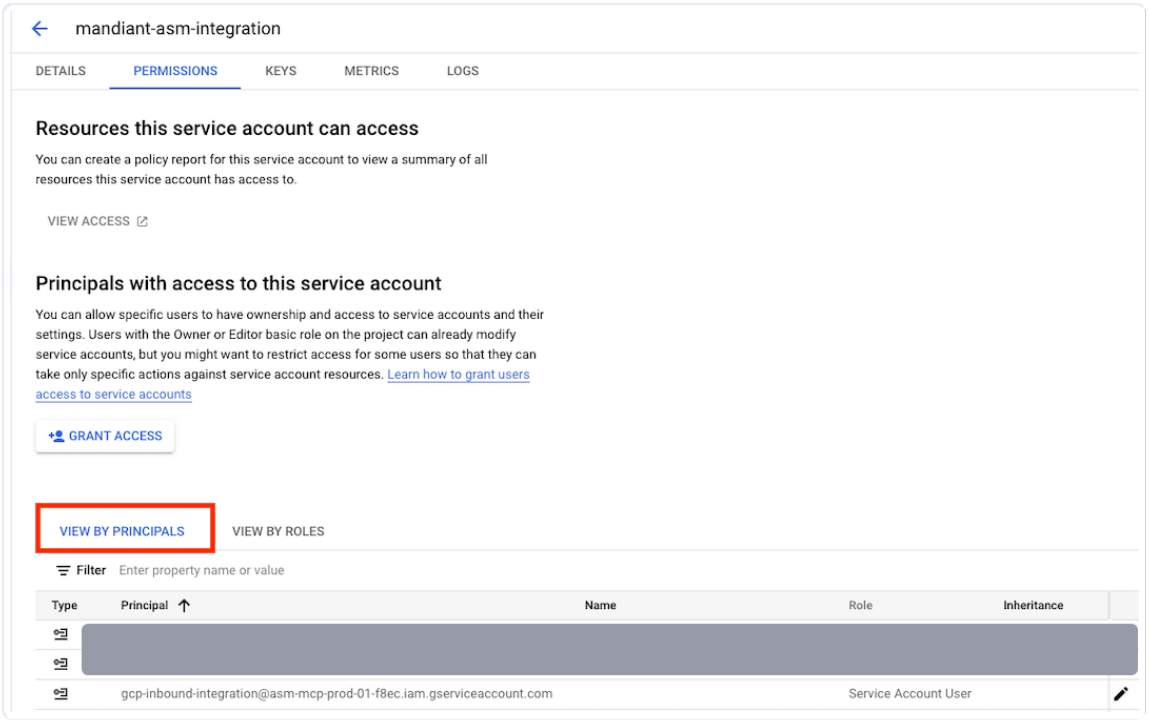


Make a note of the email belonging to your service account (not the `gcp-inbound-integration@` service account) as you'll need it in the next section, **Create Google Cloud Integration within MA-ASM**.

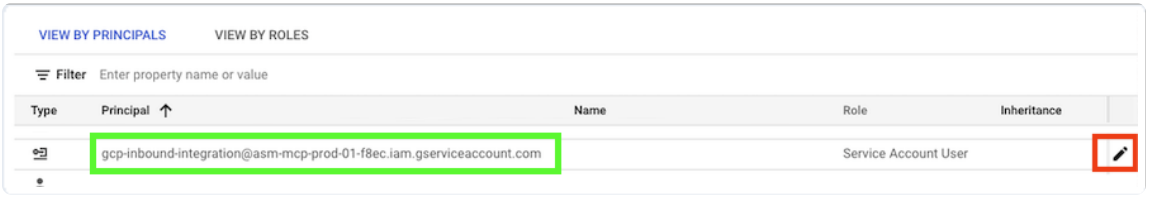
h. Click the **Permissions** tab in the navigation bar. A list of Principals appears.



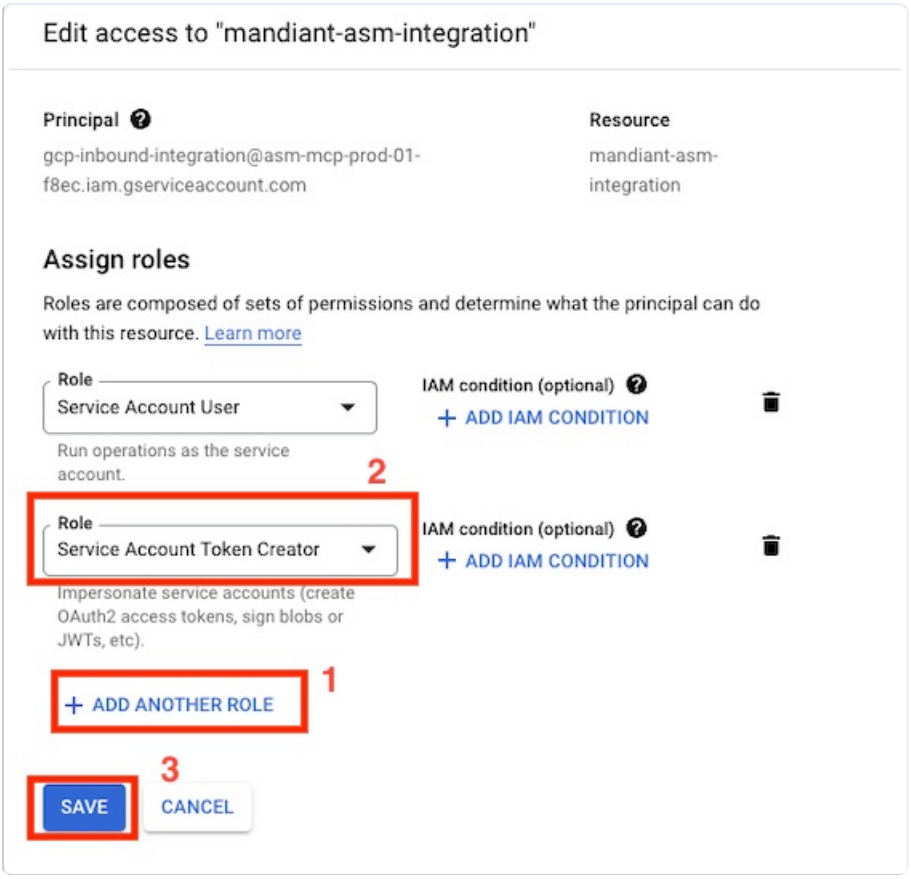
i. Ensure that the **View by principals** tab is selected and that the **Mandiant Google Cloud service account email** is listed.



j. Click the pencil icon associated with this Principal.

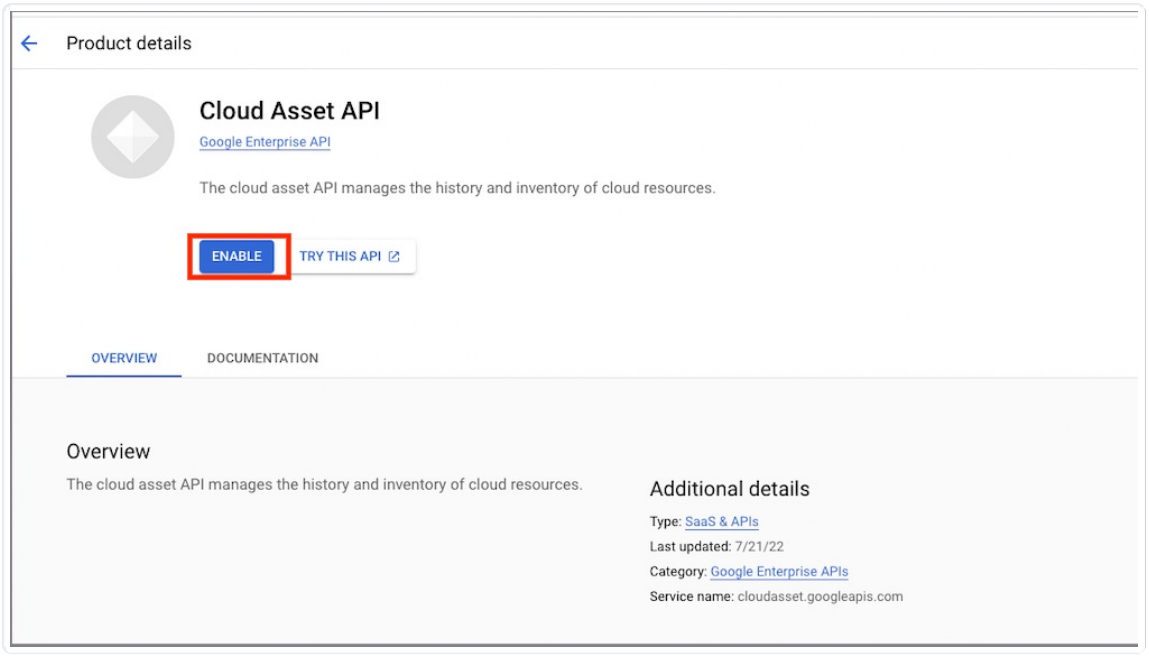


k. Under the **Assign roles** section, click **+ Add another role** and choose **Service Account Token Creator** from the drop-down menu. Click **Save**.



4. Lastly, enable the APIs for the following services by clicking each of the respective service hyperlinks and clicking **Enable**.
    - **Cloud Asset API** (https://console.cloud.google.com/marketplace/product/google/cloudasset.googleapis.com): This allows the integration to fetch resources through Cloud Asset inventory.
    - **Cloud Resource Manager API** (https://console.cloud.google.com/marketplace/product/google/cloudresourcemanager.googleapis.com): This allows the integration to fetch the list of projects.
    - **Cloud DNS API** (https://console.cloud.google.com/marketplace/product/google/dns.googleapis.com): This allows you to translate domain name requests into IP addresses.
    - **API Gateway API** (https://console.cloud.google.com/marketplace/product/google/apigateway.googleapis.com): This allows the integration to secure and manage REST APIs.

    > The `Cloud Asset API` and `Cloud Resource Manager API` must be enabled for all Google Cloud projects in-scope for MA-ASM.



Method B: Create a Service Account through gcloud CLI

1. Make sure you are authenticated with Google Cloud through the gcloud CLI by running:

   ```
   gcloud auth list
   ```

   The following output should be returned:

   ```
   Credentialed Accounts
   ACTIVE  ACCOUNT
   *       user.account@org.tld

   To set the active account, run:
     $ gcloud config set account 'ACCOUNT'
   ```

2. Set the project for which you would like the integration to fetch resources by running:

   ```
   gcloud config set project PROJECT_ID
   ```

   > *PROJECT_ID* is a variable. Be sure to use the Project ID and not the Project name.

   Successful output looks like the following:

   ```
   Updated property [core/project].
   ```

   If you see the following output, it means that the project cannot be found. This is most likely because the Project Name instead of the Project ID was used.

   ```
   WARNING: You do not appear to have access to project [REQUESTED-PROJECT-HERE] or it does not exist.
   Are you sure you wish to set property [core/project] to REQUESTED-PROJECT-HERE?

   Do you want to continue (Y/n)?
   ```

   If you would like to see all the projects in your organization and their respective Project IDs, run the following

command:

```
gcloud projects list
```

3. Create a custom role within Google Cloud that follows the principle of least privileges.
   Save the contents of the following YAML configuration:

```
title: masm-integration-role
description: integration role for the mandiant asm gcp integration
stage: GA
includedPermissions:
- cloudasset.assets.listResource
- dns.managedZones.list
- dns.resourceRecordSets.list
- resourcemanager.projects.get
- apigateway.apiconfigs.get
```

   Using the gcloud CLI, run the following command:

```
gcloud iam roles create masm_integration_role --project PROJECT_ID --file=role.yaml
```

    *PROJECT_ID* is a variable. Be sure to use the Project ID and not the Project name.

4. Create a service account.
   Using the gcloud CLI, run the following command:

```
gcloud iam service-accounts create masm-integration-svc-account --description="Service Account for MASM GCP Integration" --display-name="MASM GCP Integration Service Account"
```

    `masm-integration-svc-account` is the name of the service account created. You can use any value for this option, however, ensure to swap it in the next set of instructions.

   If successful, the following output should be returned:

```
Created service account [masm-integration-svc-account].
```

5. Bind the role created in **Step 3** to the service account created in the previous step.
   Using the gcloud CLI, run the following command:

```
gcloud projects add-iam-policy-binding PROJECT_ID --member="serviceAccount:masm-integration-svc-account@PROJECT_ID.iam.gserviceaccount.com" --role="projects/PROJECT_ID/roles/masm_integration_role"
```

    *PROJECT_ID* is a variable. Be sure to use the Project ID and not the Project name. There are three locations where this needs to be replaced.

   If successful, the following output should be returned:

```
Updated IAM policy for project [PROJECT_ID].
bindings:
- members:
  - serviceAccount:masm-integration-svc-account@PROJECT_ID.iam.gserviceaccount.com
  role: projects/PROJECT_ID/roles/masm_integration
...
```

6. Allow MA-ASM to impersonate your service account.
   Using the gcloud CLI, run the following command:

```
gcloud iam service-accounts add-iam-policy-binding masm-integration-svc-account@PROJECT_ID.iam.gserviceaccount.com --member="serviceAccount:gcp-inbound-integration@asm-mcp-prod-01-f8ec.iam.gserviceaccount.com" --role="roles/iam.serviceAccountTokenCreator"
```

> `gcp-inbound-integration@asm-mcp-prod-01-f8ec.iam.gserviceaccount.com` is the email of the service account belonging to MA-ASM.

   If successful, the following output should be returned:

```
Updated IAM policy for serviceAccount [masm-integration-svc-account@PROJECT_ID.iam.gserviceaccount.com].
bindings:
- members:
  - serviceAccount:gcp-inbound-integration@asm-mcp-prod-01-f8ec.iam.gserviceaccount.com
  role: roles/iam.serviceAccountTokenCreator
...
```

7. Enable the `Cloud Asset API`, `Cloud Resource Manager API`, `Cloud DNS API`, and `API Gateway API` services. Using the gcloud CLI, run the following commands:

```
gcloud services enable cloudresourcemanager.googleapis.com
gcloud services enable cloudasset.googleapis.com
gcloud services enable dns.googleapis.com
gcloud services enable apigateway.googleapis.com
```

   If successful, each command should return output similar to:

```
Operation "operations/acat.p2-11111111111-88a9d5b4-c262-40fa-aeee-be6029ebfef3" finished successfully.
```

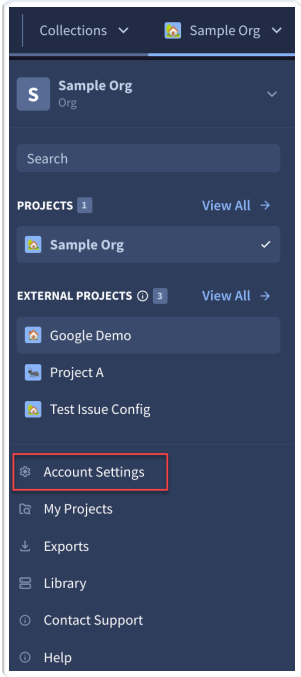
   If no response is returned, it is most likely because the service was already enabled.

> The `Cloud Asset API` and `Cloud Resource Manager API` must be enabled for all Google Cloud projects in-scope for MA-ASM.

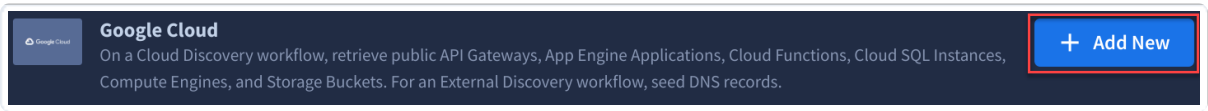
Create Google Cloud Integration within MA-ASM

The following steps take place within MA-ASM's Platform. It's assumed that a Google Cloud service account was created as documented in the **Create a Service Account** section:

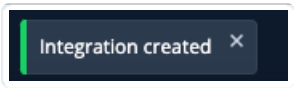
1. From the **Projects and Settings** menu in MA-ASM, select the appropriate Project then click **Account Settings**.



2. Click **Integrations**.
3. From **Inbound Integrations**, click **Add New** for Google Cloud.

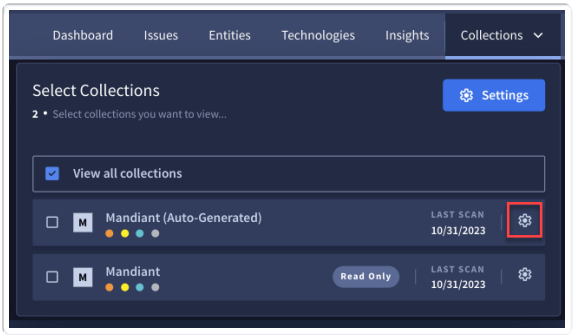


4. In the **Email** field, input the email of the Google Cloud service account, and click **Connect**.
   After successful integration, you should see an "Integration created" message:

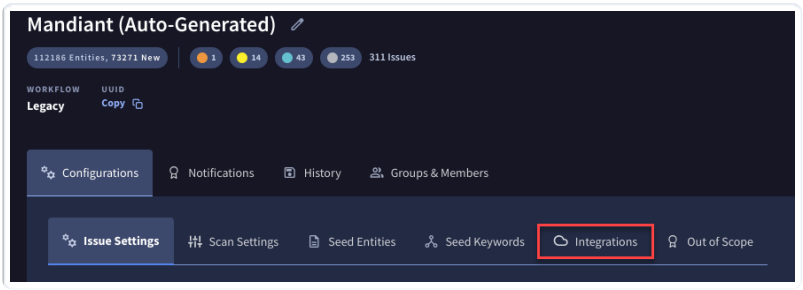


> If you see an error message, instead of an "Integration created" success message, the issue could be in how the Google Cloud service account was created. The most common reason would be to forget to grant the Google Cloud service account the **Service Account Token Creator** role.

5. Connect the integration to the appropriate Collection.
   a. Click **Collections** and click **Collection Settings** for the Collection that you want to connect the integration to.



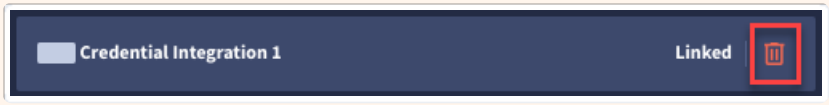
   b. Select the **Integrations** tab.



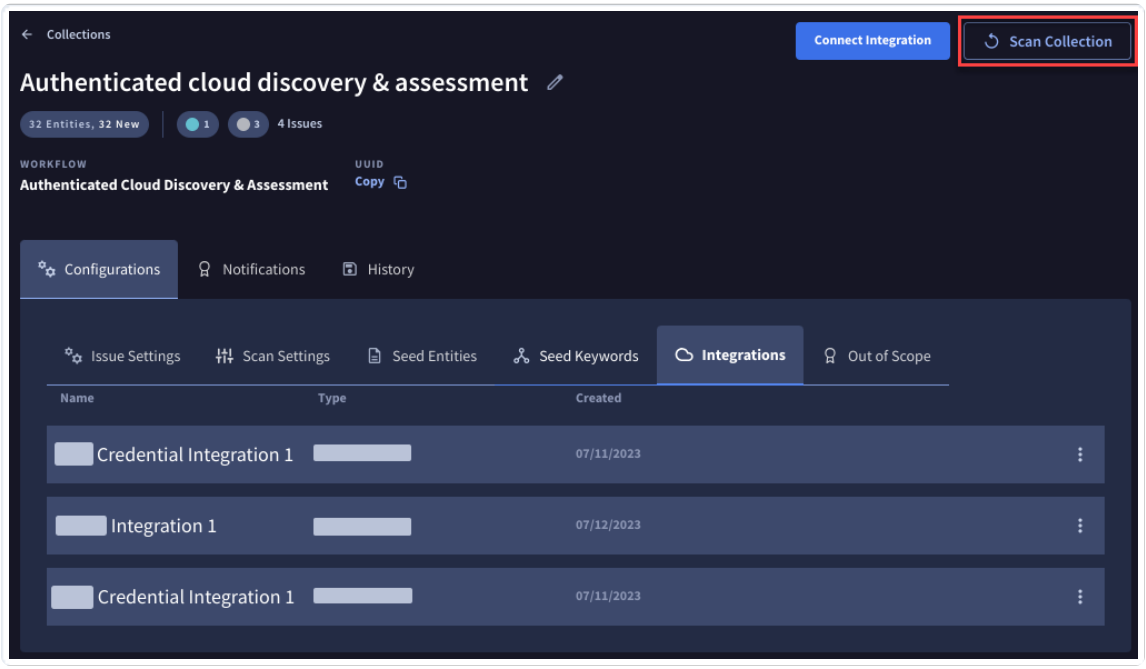
c. Select **Connect Integration** and **Link** the integration.

The integration is immediately added to the Collection.

Click to remove the integration from this Collection.



d. Click ✕ to close the **Connect Integration** pane. Click **Scan Collection** to update your Collection with the current settings and integrations. Otherwise, your newly configured integration is incorporated at your regularly scheduled scan interval.



Google Cloud Related Entities in MA-ASM

The Entities created from the Google Cloud inbound integration are as follows:

Entity Name	Google Cloud Asset Name	Relevant Raw JSON Fields	Dependencies
GcpApiGateway	API Gateway	• `name` : The hostname of the API Gateway. For example, *gateway-1337-66i59fug.uc.gateway.dev*. • `project` • `gateway_name` : Friendly name of the API Gateway. • `location` • `created_time` • `api_config` • `scoped` • `cloud_hosted`	Only API Gateways which are in the ACTIVE state are created as Entities.
GcpAppEngineApplication	App Engine Application	• `name` : apps/*project_name* • `project` • `application_hostname` • `services` : List of services running under the App Engine. • `scoped` • `cloud_hosted`	There can only be one App Engine Application per project.
GcpCloudFunction	Cloud Function	• `name` • `project` • `http_invocation_url` • `additional_info` : Information from the resource JSON. • `runtime` • `scoped` • `cloud_hosted`	Only Cloud Functions which are triggered using HTTP are created as Entities as they are deemed to be public facing. Functions configured with other triggers, such as "write to bucket," are ignored.
GcpCloudSQLInstance	Cloud SQL Instance	• `name` • `project` • `ip_addresses` • `database_version` • `region` • `cloud_hosted` • `scoped`	Only Cloud SQL Instances with public IP addresses and in the RUNNABLE state are created as Entities.

Entity Name	Google Cloud Asset Name	Relevant Raw JSON Fields	Dependencies
GcpComputeEngineInstance	Compute Engine	• `name` : *project/ name* • `project` • `zone` • `public_ip_addresses` • `cloud_hosted` • `scoped`	Only Compute Engine Instances with public IP addresses and in the RUNNING state are created as Entities.
GcpStorageBucket	Storage Bucket	• `name` • `bucket_url` • `project` • `location` • `anonymous_access_forbidden` • `scoped` • `cloud_hosted`	If `anonymous_access_forbidden` is `false` , a World Readable Google Cloud Storage Bucket Issue is created during the enrichment process.