# GCP Integration Guide

CBOM Secure · GCP Discovery Sensor

## Overview

This guide describes how to configure Google Cloud Platform (GCP) so that the CBOM Secure GCP Discovery Sensor can discover Cloud KMS asymmetric keys. It covers creating a project, enabling the required APIs, provisioning a key ring and key, creating a read-only service account, and supplying credentials to the sensor. Steps are provided for both the GCP Console and the gcloud CLI.

## Prerequisites

- A Google Cloud account with a billing account set up (the free tier is sufficient).
- Permission to create projects, service accounts, and KMS resources (Owner or an equivalent role).
- The gcloud CLI installed (optional — required only for the CLI steps).
- Python 3.x installed on the machine that runs the sensor.
- The CBOM Secure GCP Discovery Sensor package ready for configuration.

## Step-by-Step Guide

### Step 1: Create a GCP Project

- Go to https://console.cloud.google.com/.
- At the top, click the project selector dropdown, then click New Project.

```
Enter a Project name (for example, cbom) and select a billing account.
```

- Click Create, then select the new project.

### Step 2: Enable Required APIs

Navigate to https://console.cloud.google.com/apis/library and enable the following APIs for the project:

| API name | Why it is required |
|---|---|
| Cloud KMS API | Required for key management and key discovery. |
| IAM API | Required to manage service accounts and roles. |

Click Enable on each.

### Step 3: Create a Key Ring and Key

#### Using the GCP Console

- Go to Security → Key Management → Key Rings.

```
Click Create Key Ring — Name: test-key-ring, Location: global.
```

- Click Create Key with: Name test-asym-key, Key Purpose Asymmetric Sign, Algorithm RSA_SIGN_PKCS1_2048_SHA256, Protection Level Software (or HSM if available).

**Alternatively, using gcloud CLI**

```
gcloud kms keyrings create test-key-ring --location global

gcloud kms keys create test-asym-key \
  --location global \
  --keyring test-key-ring \
  --purpose asymmetric-signing \
  --default-algorithm rsa-sign-pkcs1-2048-sha256
```

### Step 4: Create a Service Account

- Go to IAM & Admin → Service Accounts.

```
Click Create Service Account — Name: cbom-kms-reader, Description: Used for CBOM GCP discovery.
```

- Click Create and Continue.
- Grant the role Cloud KMS Viewer (and, optionally, Secret Manager Viewer if you plan to add certificates later).
- Click Done.

### Step 5: Create and Download a JSON Key

- In the service account list, click the three dots → Manage keys.
- Click Add Key → Create new key.
- Select JSON and click Create.

```
Save the key file (for example, cbom-sa-key.json) securely.
```

### Step 6: Install Python Dependencies

```
pip install google-cloud-kms cryptography
```

### Step 7: Set the GOOGLE_APPLICATION_CREDENTIALS Variable

Point the sensor at the downloaded service account JSON key. Run the command for your shell:

**Command Prompt**

```
set GOOGLE_APPLICATION_CREDENTIALS=C:\CBOM-Services\creds\cbom-sa-key.json
```

**PowerShell**

```
$env:GOOGLE_APPLICATION_CREDENTIALS="C:\CBOM-Services\creds\cbom-sa-key.json"
```

**Note** This environment variable must be set in the same session that runs the sensor, unless you persist it (see Step 8).

### Step 8: Persist the Environment Variable (Optional, Windows)

- Search for Environment Variables in the Start menu.
- Under System Properties → Environment Variables, click New (in User or System variables).

```
Name: GOOGLE_APPLICATION_CREDENTIALS; Value: the full path to the JSON key.
```

- Click OK, apply the changes, then restart Command Prompt or the system.

### Step 9: Configure the CBOM Secure GCP Sensor

Create the sensor configuration file:

```
ExtName: Discover_GCPKMS
project_id: "cbom-468113"
location_id: "global"
key_ring_id: "test-key-ring"
```

### Step 10: Validate Discovery

Rerun the CBOM Secure GCP discovery task and confirm that the Cloud KMS keys in the configured key ring are discovered without authentication or permission errors.