Google Cloud External Key Manager and Entrust KeyControl Vault

Integration Guide

2024-11-21

© 2025 Entrust Corporation. All rights reserved.

Table of Contents

Chapter 1. Introduction

This document describes the integration of Google Cloud Platform (GCP) External Key Manager (EKM), referred to as GCP EKM in this guide, with the Entrust KeyControl Vault Key Management Solution (KMS).

1.1. Documents to read first

This guide describes how to configure KeyControl Vault server as a KMS in GCP. To install and configure the KeyControl Vault server see KeyControl Vault Installation and Upgrade Guide.

Also refer to the documentation and set-up process for GCP EKM in the Google Cloud External Key Manager documentation.

1.2. Product configurations

Entrust has successfully tested the integration of KeyControl Vault with GCP EKM in the following configurations:

System	Version
KeyControl Vault	10.2 / 10.3.0

1.3. Features tested

Entrust has successfully tested the following features:

Feature	Tested
Create cloud key	✓
Enable cloud key	✓
Disable cloud key	✓
Rotate cloud key	✓
Delete a cloud key	✓
Cancel cloud key deletion	✓

Feature	Tested
Access an object protected by cloud key in GCP	✓
Sign/Verify an input file with GCP cloud key	✓

1.4. Requirements

Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

Chapter 2. Install and configure KeyControl Vault

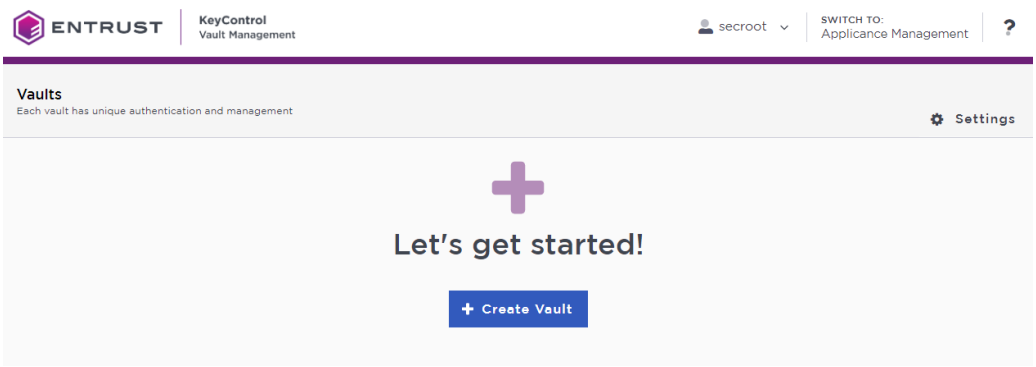
2.1. Deploy a KeyControl Vault cluster

For this integration, KeyControl Vault was deployed as a two-node cluster.

Follow the installation and set-up instructions in KeyControl Vault Installation and Upgrade Guide.

2.2. Create a KeyControl Cloud Key Management Vault

1. Sign in to the KeyControl Vault Manager.
2. In the home page, select **Create Vault**.

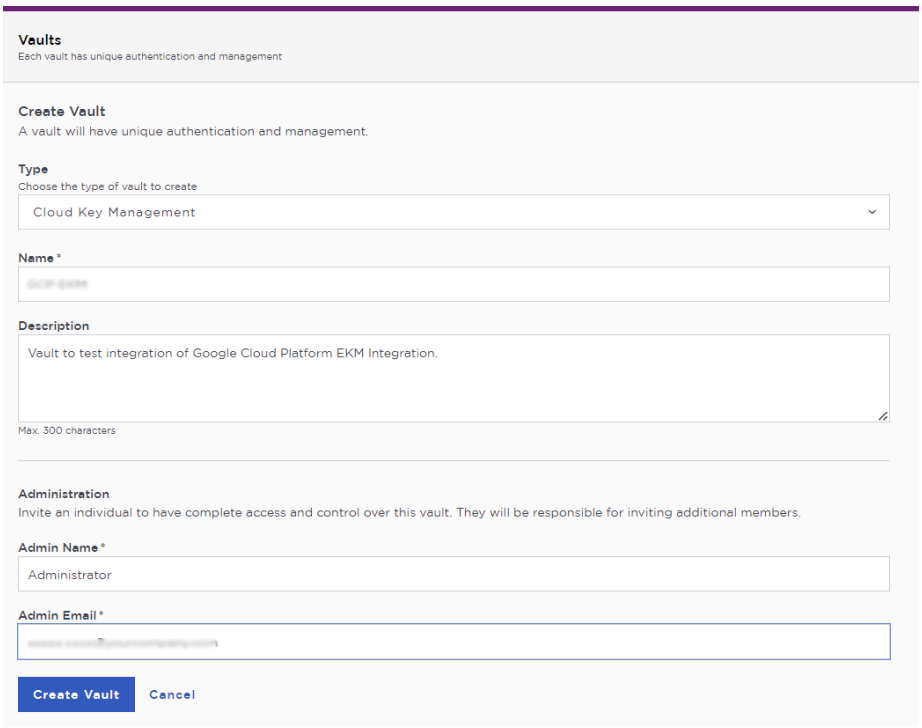


3. Select **Create Vault**.

   The **Create Vault** dialog appears.

4. In the **Type** drop-down box, select **Cloud Key Management**. Enter the required information.
5. Select **Create Vault**.

   For example:



6. When you receive an email with a URL and sign-in credentials to the KeyControl vault, bookmark the URL and save the credentials.

   You can also copy the sign-in credentials when the vault details gets displayed and use that to sign in using the vault URL.

7. Sign in to the URL provided.

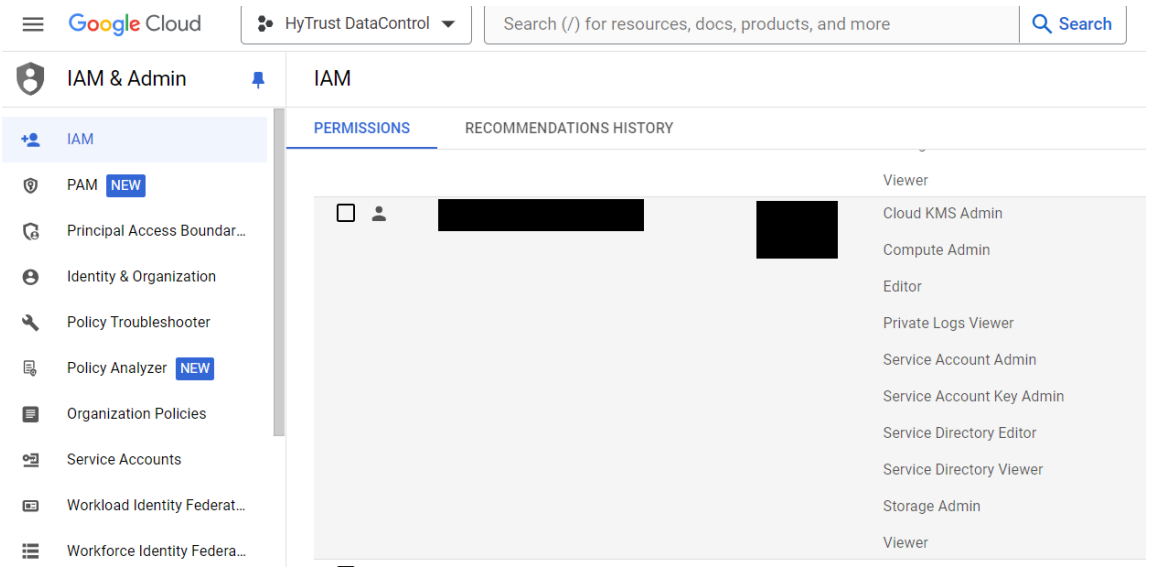
   Change the initial password when prompted.

Chapter 3. Configure Google Cloud Platform

3.1. Required GCP permissions

The GCP account performing this integration had the following permissions. These were granted by the project admin. Not all these permissions are required to perform this integration.

- Cloud Build Editor
- Cloud KMS Admin
- Compute Admin
- Deployment Manager Editor
- Private Logs Viewer
- Service Account Admin
- Service Account Key Admin
- Service Account User
- Service Management Administrator
- Service Usage Admin
- Storage Admin
- Viewer



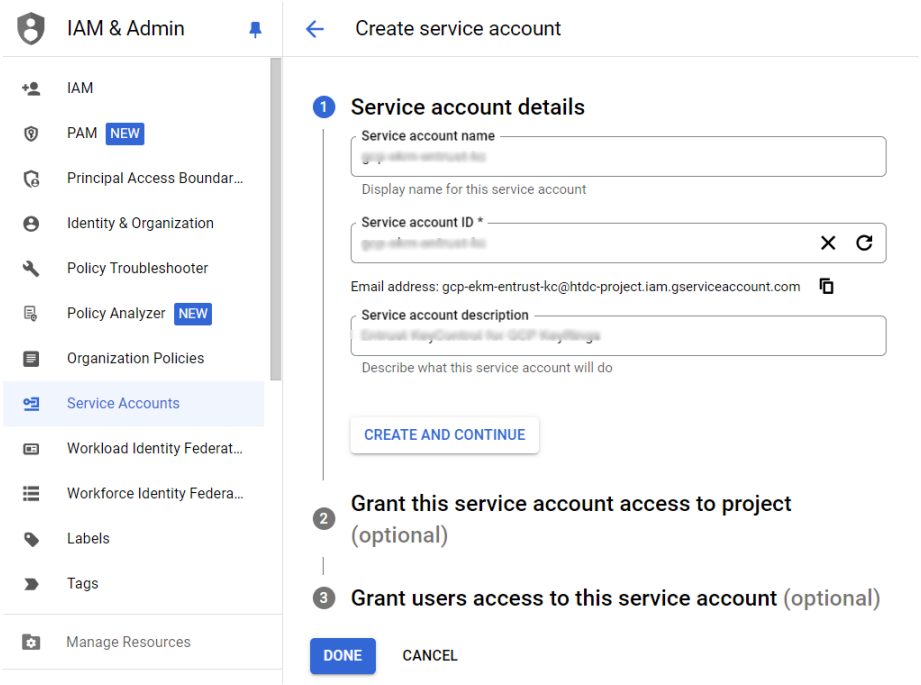
3.2. Create a service account in GCP

A service account needs to be created in a GCP IAM. This service account will be used by

KeyControl Vault to access the GCP key rings. Once created, this service account needs permissions that have to be granted by the project admin.

1. Open a browser and sign in to the GCP portal https://console.cloud.google.com.
2. Select **IAM & Admin** on Google Cloud Menu.
3. Select **Service Accounts** in the left-hand pane.
4. Select **CREATE SERVICE ACCOUNT**.
5. Enter the **Service account details**.

   For example:



   - Select **CREATE AND CONTINUE**
   - In the **Grant this service account access to project** section, select **Continue**.
   - In the **Grant users access to this service account** section, select **DONE**.

3.3. Service Account Permissions

1. Open a browser and sign in to the GCP portal https://console.cloud.google.com.
2. Select **IAM & Admin** on Google Cloud Menu.
3. Select **Service Accounts** in the left-hand pane.
4. Select the service account that you have just created.
5. In the **DETAILS** tab.
   a. Take note of the **Account Name**.

b. Take note of the **Unique ID**.

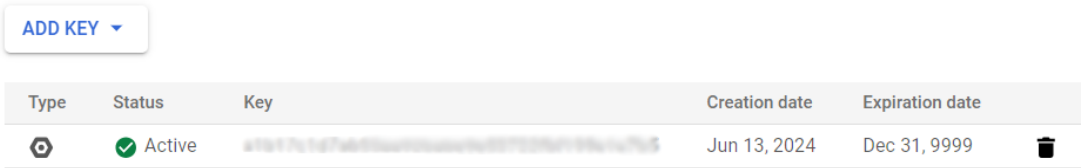
6. The following roles were given to this service account by the system admin after it was created:
    - Browser
    - Cloud KMS Admin
    - Service Account Key Admin

3.4. Create a key for the service account

A key needs to be created for the service account created in Create a service account in GCP. This key will be used by KeyControl Vault to access the GCP service account.

1. Open a browser and sign in to the GCP portal: https://console.cloud.google.com.
2. Select **IAM & Admin** on Google Cloud Menu.
3. Select **Service Accounts** in the left-hand pane.
4. Select the service account created in Create a service account in GCP from the list.
5. Select the **KEYS** tab.
6. Select **ADD KEY** and then select **Create new key**.
7. Select **JSON** from the available **Key type** options.
8. Select **CREATE**. A pop-up message appears indicating that the key created was downloaded to your computer.
9. Verify by checking your `Downloads` folder that a **.json** file was created in the Downloads folder.
10. Take note of the new key in the GCP console.

    For example:



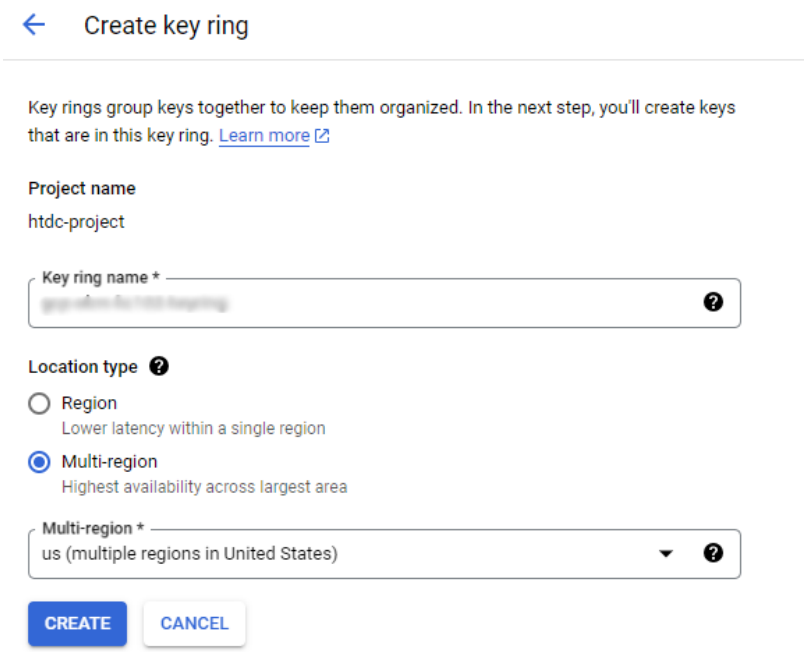
3.5. Create a GCP key ring

This key ring will be used to store keys managed by KeyControl Vault. A new GCP key ring was created for this integration to show the entire process. You can use an existing key ring instead.

If you are using an existing GCP key ring, proceed to section Create a KeyControl Vault CSP account for the GCP service account directly, skipping this section entirely.

1. Open a browser and sign in to the GCP portal: https://console.cloud.google.com.
2. In the navigation menu select **Security** > **Key Management**.
3. Select **+ CREATE KEY RING**.
4. Enter the **Key ring name** and select the **Location type**.

   For example:



5. Select **CREATE** to create the key ring
6. Select **CANCEL** in the **Create key** pane.

Chapter 4. Configure KeyControl as GCP KMS

4.1. Create a KeyControl Vault CSP account for the GCP service account

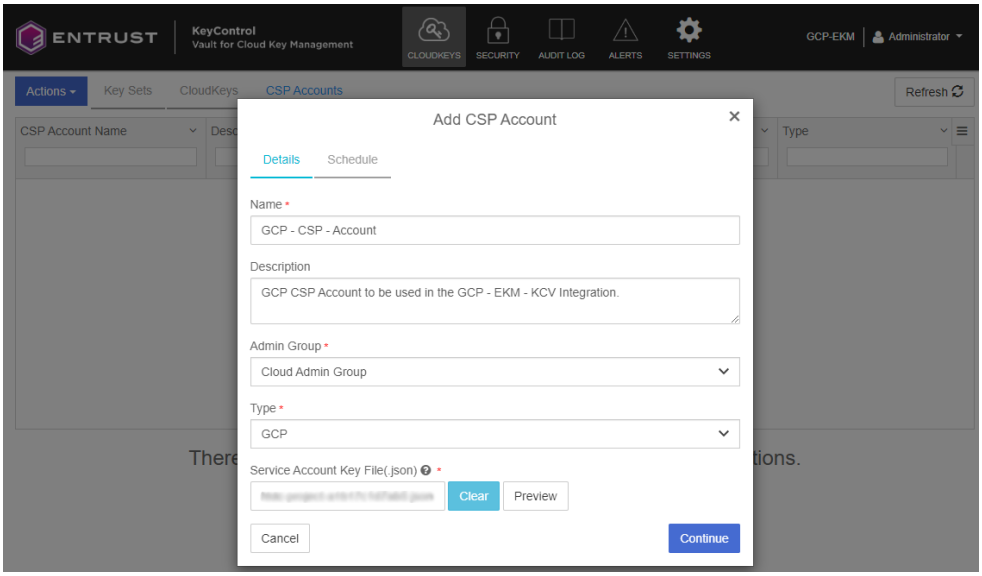
The following steps establish the connection between KeyControl Vault and GCP, making KeyControl Vault the CSP of the GCP service account. You must have created a service account in GCP and downloaded the JSON file before you can add a CSP account. For more information see GCP Service Account Requirements.

1. Log into the KeyControl Cloud Key Management Vault webGUI using an account with Cloud Admin privileges.
2. Select the **CLOUDKEYS** icon on the toolbar.
3. Select the **CSP Accounts** tab.
4. Select the **Action** icon and then **Add CSP Account** from the drop-down menu that appears.

   The **Add CSP Account** dialog appears.

5. In the **Details** tab:
   a. Enter the **Name** and **Description**.
   b. From the **Admin Group** drop-down menu box, select **Cloud Admin Group**.
   c. From the **Type** drop-down menu box, select **GCP**.
   d. In the **Service Account Key File (.json)** field, select the file download to your computer in Create a key for the service account.

      For example:



6. Select **Continue**.
7. In the **Schedule** tab, select **Never**.
8. Select **Add**.

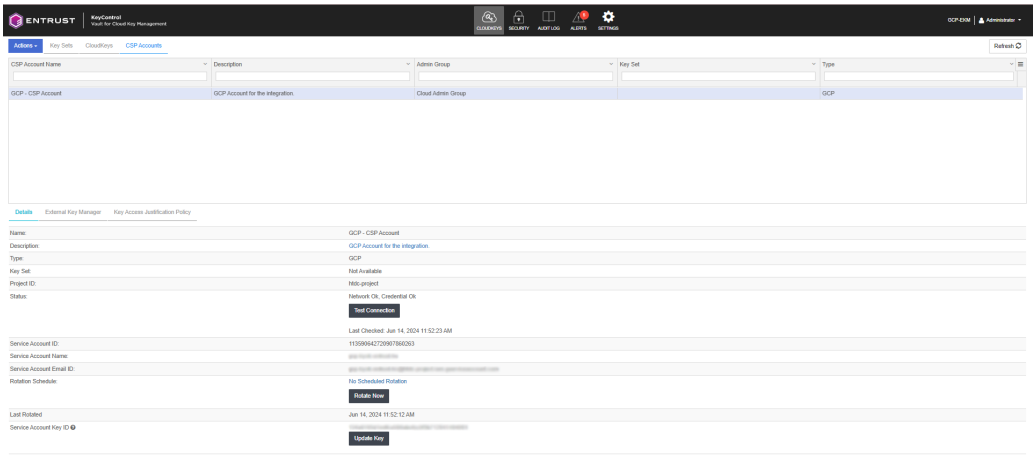
When the service account keys are rotated, the KeyControl Cloud Key Management Vault creates a new key and replaces the key that was used when you registered the CSP account. Do not delete the service account key.

4.2. Update the EKM, Key Access Justification Policy, and EKM Access Control List sections

Before you can use KeyControl Vault as a GCP EKM provider, you must set up the following parameters in the **CSP Account details** page.

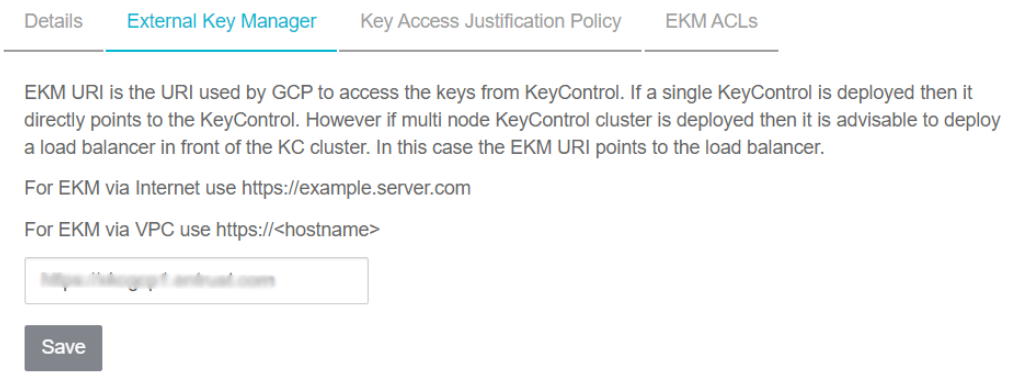
- External Key Manager (EKM URI)
- Key Access Justification Policy (optional)
- EKM Access Control List (optional)

1. View the **Details** tab on the CSP account that you created.

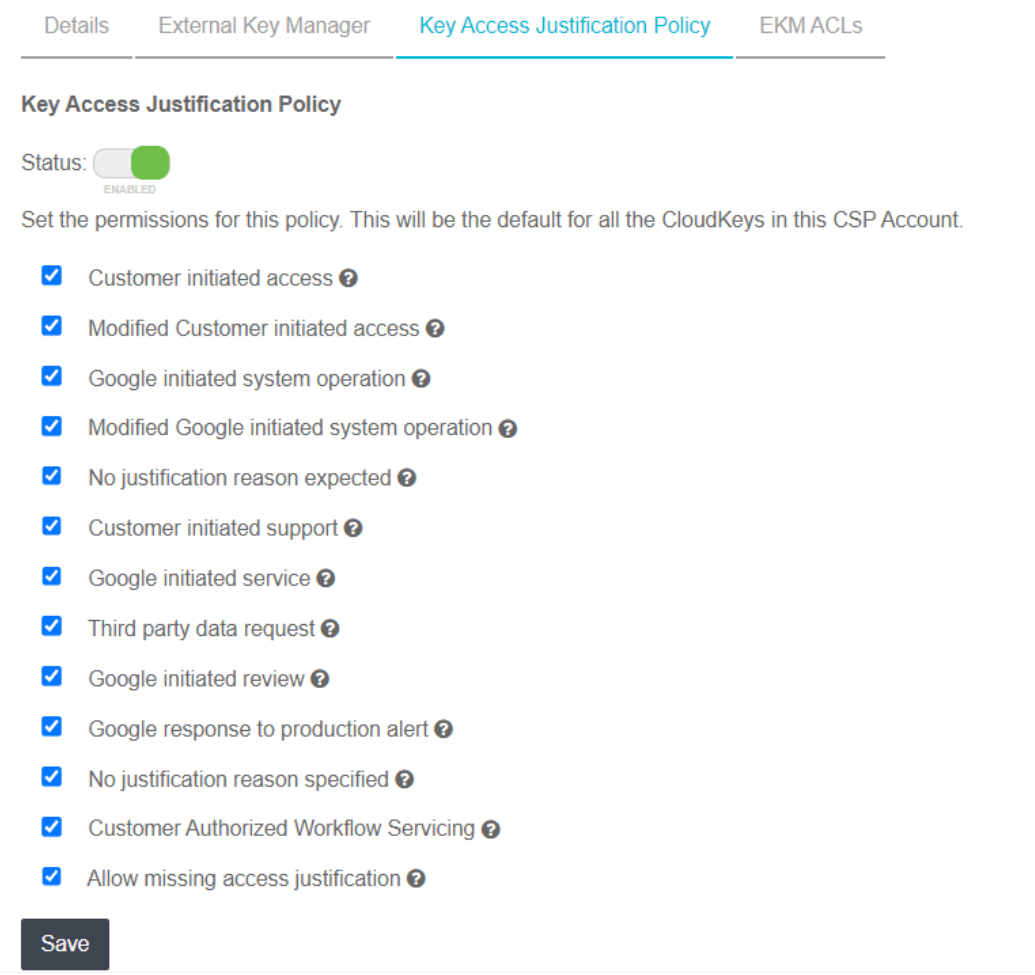


2. Select the **External Key Manager** tab and enter the URI for the KeyControl Vault server.

   Keep in mind that the EKM URI is the URI used by GCP to access the keys from KeyControl Vault. If a single KeyControl Vault is deployed, then it directly points to that vault. However if you deploy a KeyControl Vault cluster with multiple nodes, then we recommend deploying a load balancer in front of the KeyControl Vault cluster. In this case the EKM URI points to the load balancer.



3. Select **Save**.
4. Select the **Key Access Justification Tab**. We highly recommend that you create a Key Access Justification policy at the CSP and KeySet level that specifies access justification reasons. These apply to all the keys created in this keyset, unless the policy is specified at the key level, which overrides this policy.



5. Select **Save**.
6. (Optional)

   Select the **EKM ACLs** tab to configure the EKM access control list.

   This policy also applies to all the keys which are created in the associated KeySet (to this CSP account) unless a policy at Key level overrides it. For GCP Control plane access so-called *coordinated keys*, only the CSP level permissions apply. The EKM ACL specifies the list of GCP identities and permissions they have. The identities can be specified with their service account email, that is:

   `xxxx@htdc-project.iam.gserviceaccount.com`

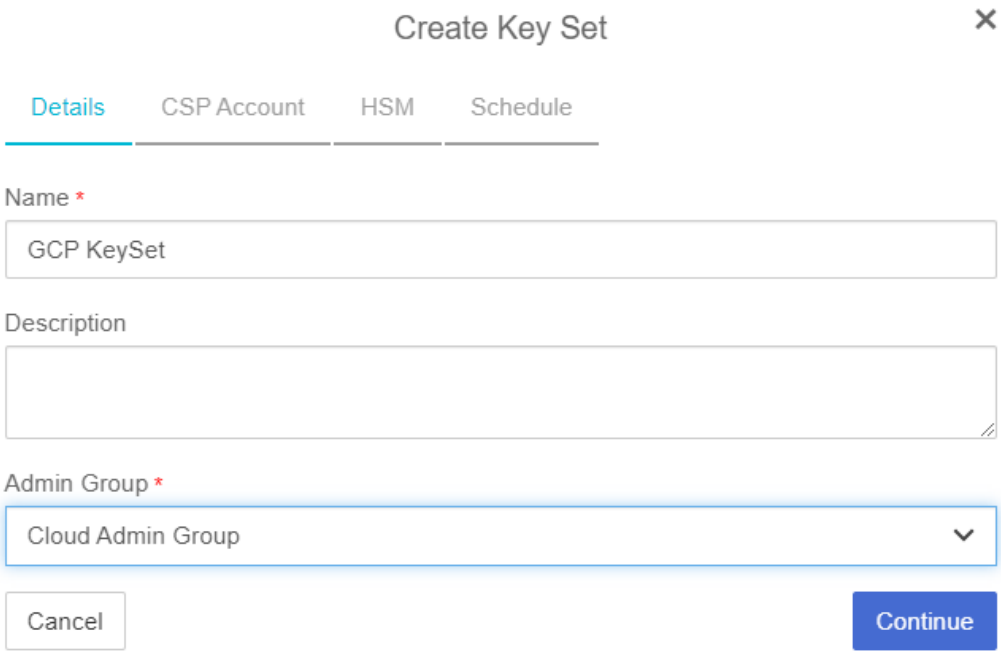
   The supported permissions are:

   - `wrap`
   - `unwrap`
   - `asymmetricSign`
   - `getPublicKey`

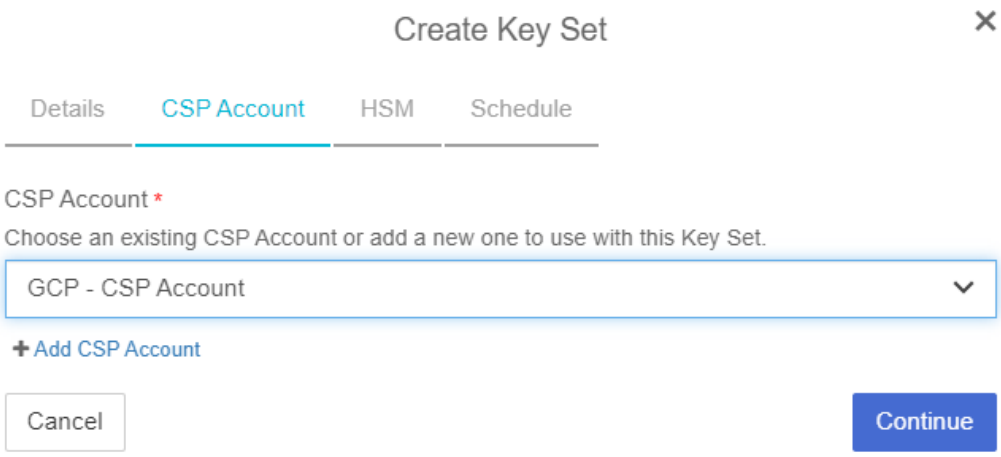
- `checkCryptoSpacePermissions`
- `createKey`
- `destroyKey`

4.3. Create a KeySet for GCP

1. Log into the KeyControl Cloud Key Management Vault webGUI using an account with Cloud Admin privileges.
2. In the top menu bar, select **CloudKeys**.
3. Select the **Key Sets** tab.
4. Select **Actions** > **Create Key Set**.
5. Select the type of key to be contained in the Key Set: **GCP Key**
6. On the **Details** tab of the **Create Key Set** dialog box, enter the following:
   a. **Name***- Enter the name for the Key Set.
   b. **Description** - Enter the optional description for the Key Set.
   c. **Admin Group** - Select the Admin Group: **Cloud Admin Group**
7. Select **Continue**.



8. On the **CSP Account** tab, select an existing CSP Account or add a new account to use with this Key Set. Select the account you created earlier.



9. Select **Continue**.
10. On the **HSM** tab, check the **Enable HSM** checkbox if you plan to use an HSM to create CloudKeys that can be uploaded to the cloud.

    When the key material is in the KMS, the HSM is no longer required. However, if you remove the CloudKey from the cloud, you will need to use the HSM to upload the key again.

    If you selected Enable HSM, select **Verify HSM** connection to test the connectivity and suitability of the configured HSM. KeyControl Vault checks if the HSM is accessible and if it supports the creation and export of relevant keys.

    Some HSM servers with old version of firmware do not support key creation and wrapping. If the connection test fails, check the firmware version of the HSM server. If it is old, update it to the latest version.

11. Select **Continue**.
12. On the **Schedule** tab, determine the default rotation schedule for the CloudKeys created in this Key Set.

    This rotation schedule is applied to all CloudKeys created in the Key Set, unless a different value is explicitly selected. If there are existing CloudKeys in the Key Set, you can update the rotation schedule of the CloudKeys to align with your selected rotation schedule by checking Apply to all CloudKeys.

13. Select **Apply**.

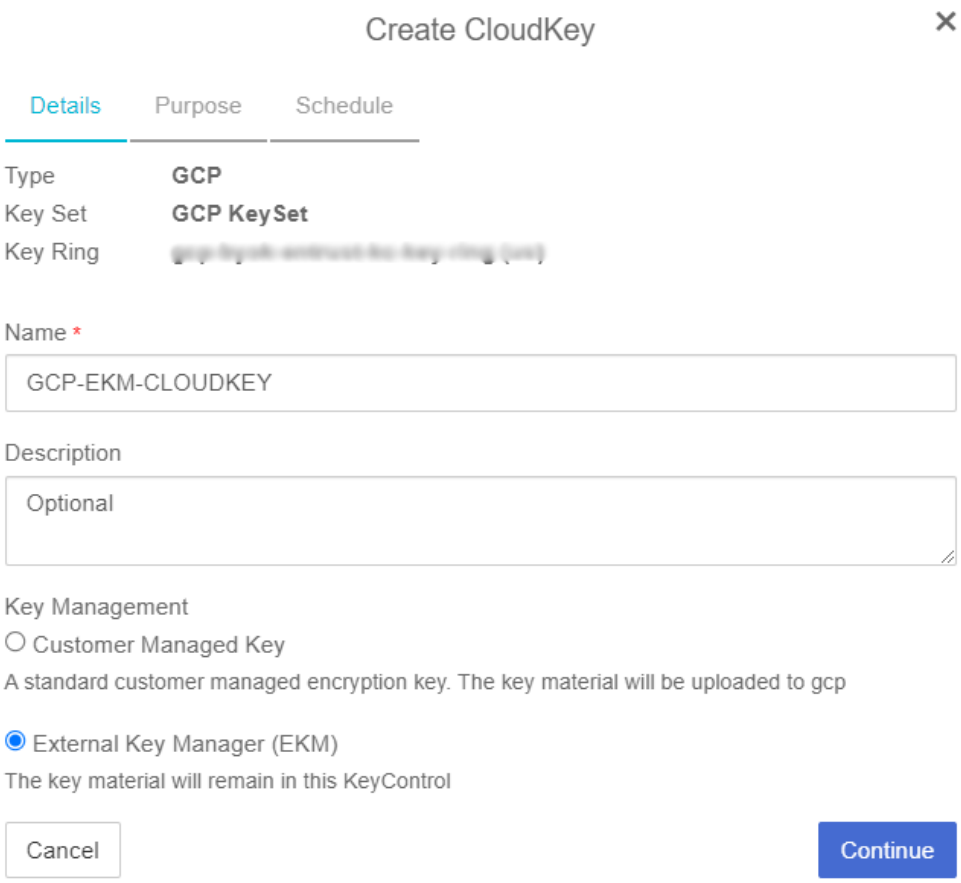
4.4. Create a CloudKey for GCP

Before you can create a CloudKey, you must Create a KeySet for GCP. This procedure is for creating an External Key Manager (EKM) key.

1. Log into the KeyControl Cloud Key Management Vault webGUI using an account with Cloud Admin privileges.
2. In the top menu bar, select **CloudKeys**.
3. Select the **CloudKeys** tab and select the **Key Set** and the **Key Ring**.

   If you do not finish the selections on the **CloudKeys** page, you will need to add them on the Details tab of the Create CloudKey dialog box.

4. Select **Actions** > **Create CloudKey**.
5. On the Details tab of the Create CloudKey dialog box, enter the following:
   a. **Name**: Enter the name for the CloudKey.
   b. **Description**: Enter the optional description for the CloudKey
   c. **Key Management**: Select **External Key Management (EKM)**.



6. Select **Continue**.
7. On the **Purpose** tab, complete the following:
   a. **Connection Type**: Select how the external key manager will be reached.
   b. **Purpose**: This can be one of the following:
      i. Symmetric encrypt/decrypt

ii. Asymmetric Sign

c. **Algorithm**: Select the algorithm that matches the purpose you selected. This can be one of the following:

i. For Symmetric encrypt/decrypt: **External symmetric key**

ii. For Asymmetric Sign:

A. Elliptic Curve P-256 - SHA256 Digest

B. Elliptic Curve P-384 - SHA384 Digest

C. 2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest

D. 3072 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest

E. 4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest

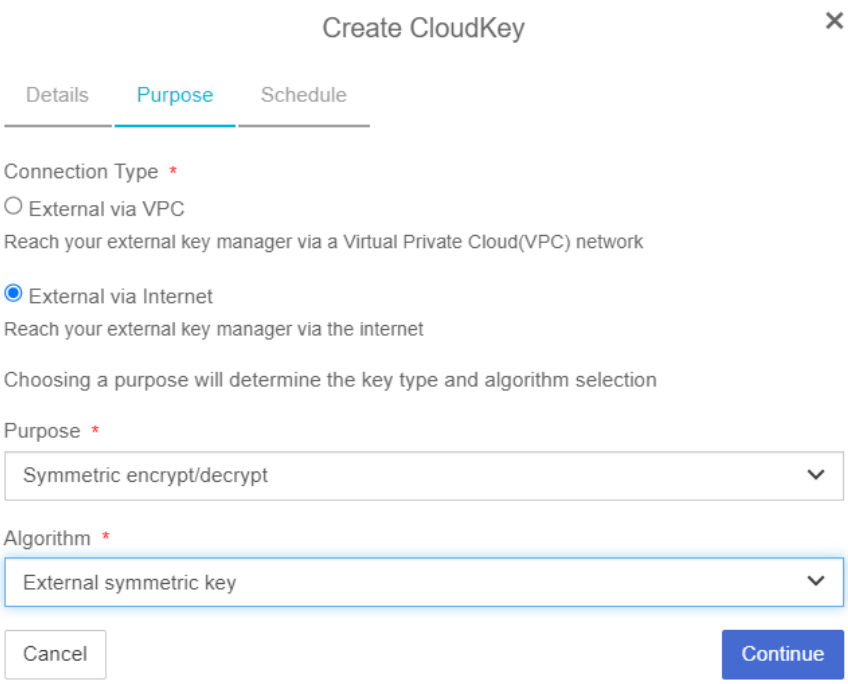
F. 4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest

G. 2048 bit RSA - PSS Padding - SHA256 Digest

H. 3072 bit RSA - PSS Padding - SHA256 Digest

I. 4096 bit RSA - PSS Padding - SHA256 Digest

J. 4096 bit RSA - PSS Padding - SHA512 Digest

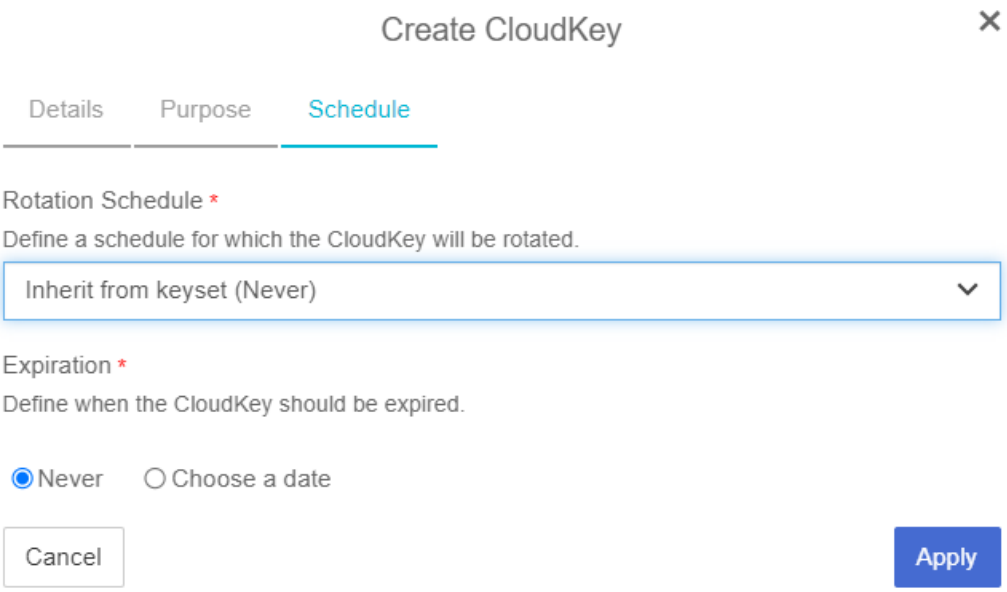


8. Select **Continue**.
9. On the Schedule tab, determine the rotation schedule for the CloudKey. This can be one of the following:

a. Inherit from Key Set—The - CloudKey will use the default schedule from the Key Set. If the Key Set schedule changes after the CloudKey is created, the CloudKey schedule will not be updated.

b. Never — The CloudKey will never be rotated.
c. Once a year — The CloudKey will be rotated once a year.
d. Every 6 months — The CloudKey will be rotated once every 6 months.
e. Every 30 days — The CloudKey will be rotated once every 30 days.
f. Other — The CloudKey will be rotated at the interval you select.

10. Select when the CloudKey should expire. This can be `Never`, or you can select a specific date.



If you selected an expiration date, select the Expire Action to define what happens to the CloudKey when it expires.

When the CloudKey expires, the selected Expire Action is performed on the key. The KeyControl Vault handles the expiration date and expire action. The expire date is not set in the cloud service provider.

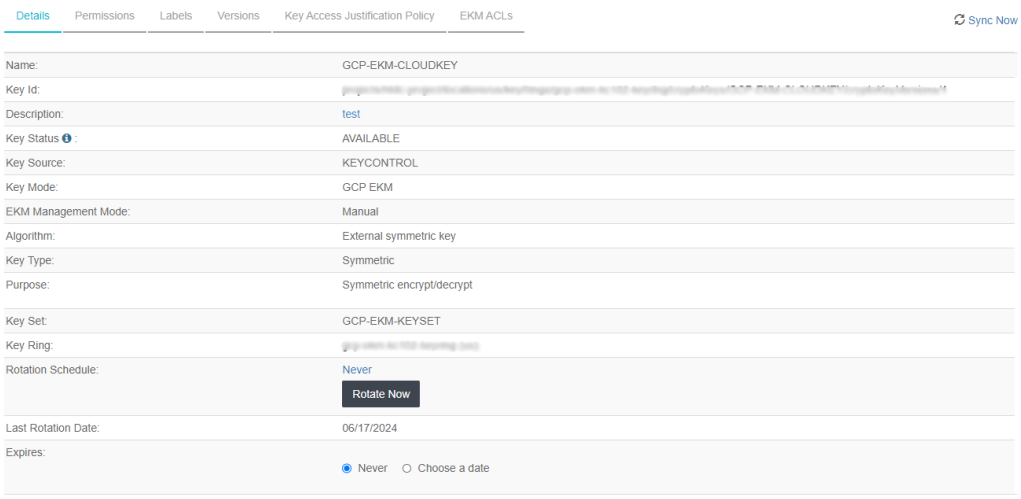
11. Select **Apply**.

If you get errors about not being able to validate the TLS server certificate for the key, you will need to install the SSL certificate in the KeyControl Vault node.

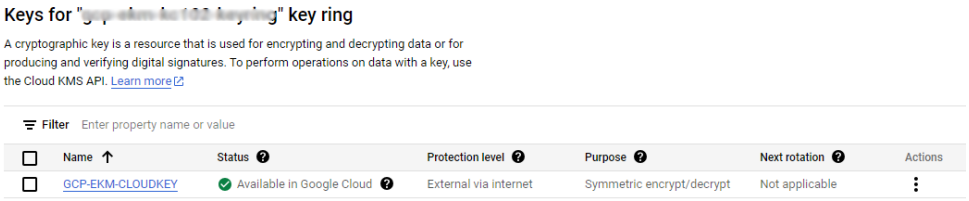
4.5. Verify the CloudKey

You need to make sure the CloudKey status is **AVAILABLE** so it can be used in GCP.

1. Verify the Key is Available, by selecting the CloudKey and checking its status in the details tab.



2. Verify the Key status in the GCP Platform.
   a. Open a browser and sign in to the GCP portal: https://console.cloud.google.com.
   b. In the navigation menu select **Security** > **Key Management**.
   c. Look for the key ring that was used in the CloudKey creation process in KCV and select on it to view its details. You should see the CloudKey you created. Its protection level should state **External Via Internet**.



Chapter 5. Test integration

5.1. Check if Cloud Key is Working as expected

To check if the Cloud Key created earlier is functioning as designed we will do the following:

- Create a Cloud Storage Bucket
- Test if items in the cloud storage bucket are protected by the Cloud Key

5.2. Create a Cloud Storage Bucket

This bucket will be used to test the Cloud Key when attempting to view an object stored in the bucket. All objects will be encrypted by the cloud key, and only visible when the cloud key is active. If the cloud key has been disabled, the object in the bucket will not be accessible.

1. Copy the Cloud Key name created earlier.
2. Open a browser and sign in to the GCP portal: https://console.cloud.google.com.
3. In the navigation menu select **Cloud Storage** > **Buckets**.
4. Select **Create**.
5. Enter the **Name** of the bucket.
6. Select **Continue**.
7. On the **Choose Where to Store Your Data**, for **Location Type**, select **Region**.
8. Select **us-east1**
9. Select **Continue**.
10. On the **Choose a storage class for your data**, select **Set as a default class** and select **Standard**.
11. Select **Continue**.
12. On the **Choose how to control your objects**, deselect **Enforce public access prevention on this bucket**.
13. Select **Continue**.
14. On the **Choose how to protect your data**, Expand the **Data Encryption** section and select **Cloud KMS Key**.
    a. For **Key Type**, select **Cloud KMS**.
    b. For **Select a customer-managed key**, enter the cloud key name you created.

15. Select **Create**.

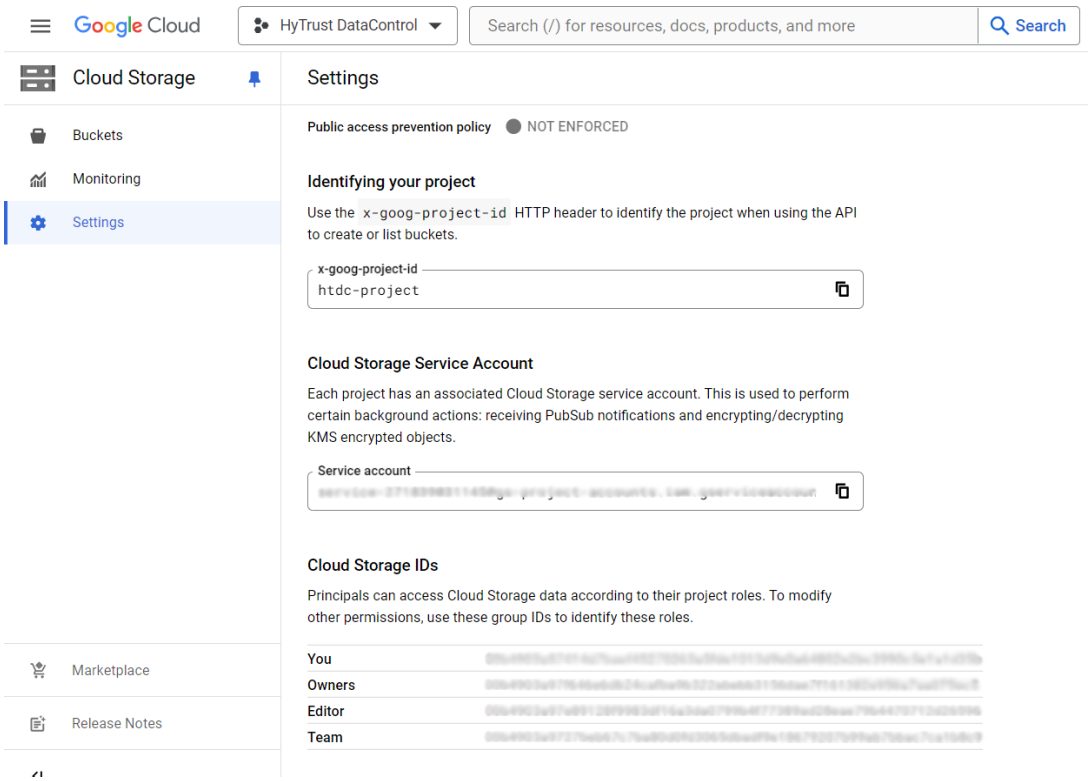
Possible issues while creating the bucket

If the service agent account for the Cloud Storage service does not have the appropriate roles, the creation of the bucket may fail with the following error:

> You do not have sufficient permissions for Google Cloud to display whether the correct IAM policy for the specified encryption key exists. Check that your Cloud Platform project's service account, `service-271839031145@gs-project-accounts.iam.gserviceaccount.com`, has the `cloudkms.cryptoKeyEncrypterDecrypter` role for the specified key. Without this role, you may not be able to encrypt or decrypt objects using the key which will prevent you from uploading or downloading objects.

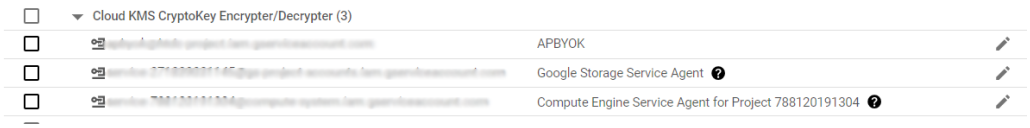
Talk to your GCP administrator and ask for the role to be added to the service account.

The service account above is a default "Service Agent" of the Cloud Storage service and this Agent doesn't have access to role `cloudkms.cryptoKeyEncrypterDecrypter`. If you select **Settings** under **Cloud Storage**, you will be able to see the service account being used.



Once the role has been granted, you can see it.

1. In the navigation menu select **IAM & Admin**.
2. Make sure **Include Google-provided role grants** is selected.
3. Select the **View by Roles** tab.
4. Look for the **Cloud KMS CryptoKey Encrypter/Decrypter (x)** Role and you should be able to see the account listed under that role.

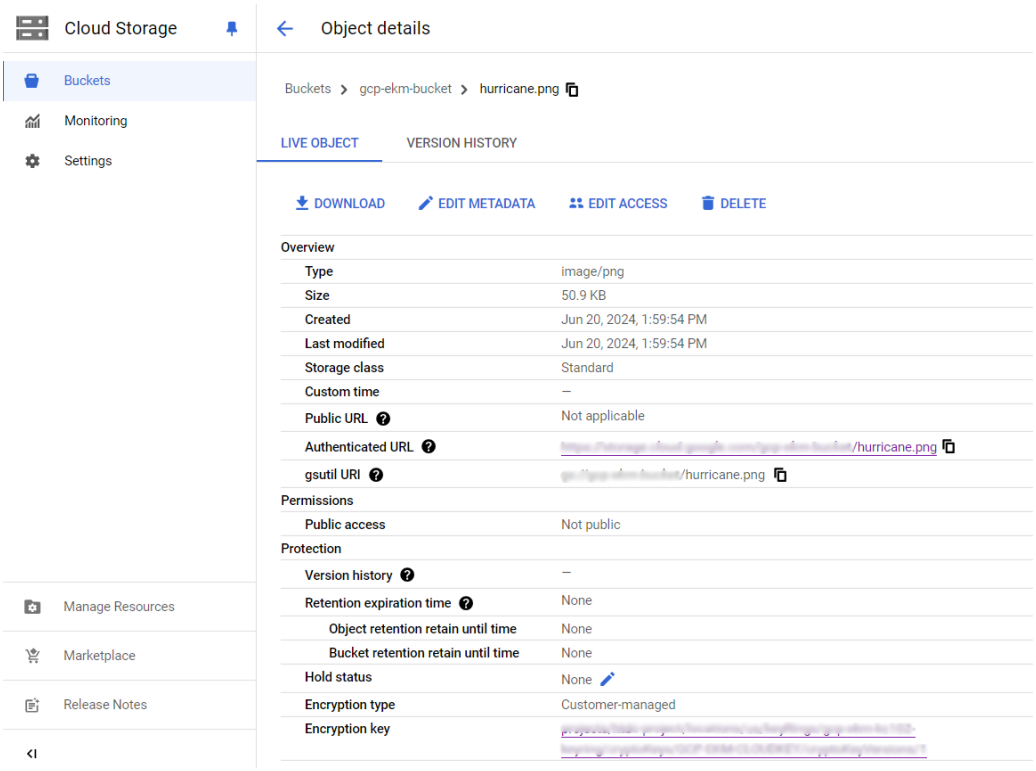


5. Now go back and attempt to create the bucket again.

   You may see the message again, but this time GCP will create the bucket which will allow you to do the testing.

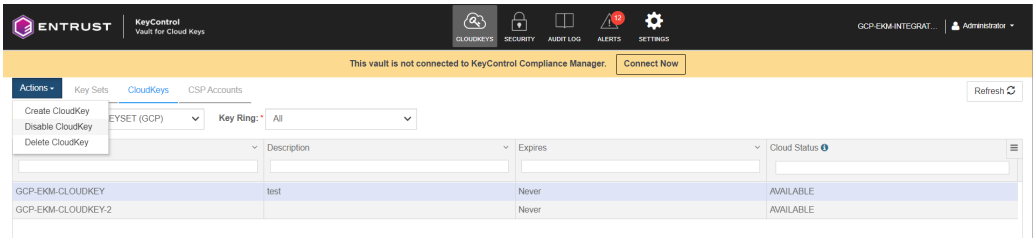
5.3. Test access to an object in the bucket

1. Upload a file to the bucket. We suggest you upload an image.
2. Once the file has been uploaded, select on it to see its details. For example:



3. Copy the **Authenticated URL** in the **Object Details** view and use it in another tab in the browser. You should be able to see the contents of the file. In this case an image:

4. Now go back to the KeyControl Vault and disable the cloudkey.
5. Log into the KeyControl Cloud Key Management Vault webGUI using an account with Cloud Admin privileges.
6. In the top menu bar, select **CloudKeys**.
7. Select the **CloudKeys** tab and select the **Key Set**.
8. Select the CloudKey used in the Bucket in GCP. In the **Actions** menu, select **Disable CloudKey**.



9. Now if you try to access the image file uploaded in the bucket earlier, you should see the following message:

> The Cloud Storage service agent does not have permission to access the KMS key in Cloud EKM. Grant the appropriate permissions in your external key manager.

10. Go back to the KeyControl Vault and enable the cloudkey.
11. Select the CloudKey used in the Bucket in GCP. In the **Actions** menu, select **Enable CloudKey**.
12. The image file now is visible again.

5.4. Rotate a cloud key in KeyControl

To rotate a cloud key in KeyControl:

1. Sign in to the KeyControl Vault URL bookmark from Create a KeyControl Cloud Key Management Vault.
2. Select the **CLOUDKEYS** icon on the toolbar.
3. Select the **CloudKeys** tab.
4. From the **Key Set** menu, select the **Key Set** created in Create a KeySet for GCP.
5. From the **Key Ring** menu, select the key ring created in Create a GCP key ring.
6. Select the key to rotate.
7. Select **Rotate Now** in the **Details** tab of the key.
8. Once rotated, select the **Versions** tab of the key, to see that a new version of the key has been created.
9. In GCP, navigate to **Security** > **Key Management**.
10. In the **KEY RINGS** tab, select the key ring created in Create a GCP key ring.
11. Select the key you just rotated in KeyControl.
12. Verify that the key has been rotated in GCP in synchronization with KeyControl.

5.5. Delete a cloud key in KeyControl

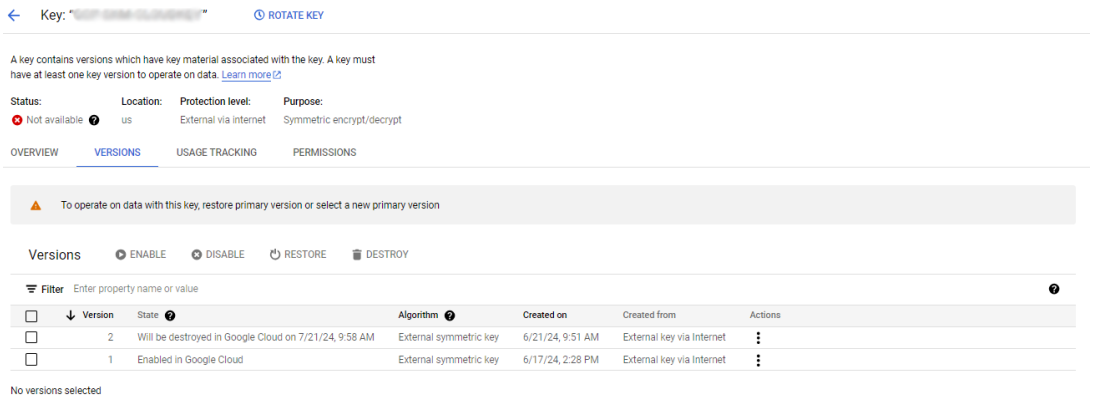
A deleted cloud key in KeyControl will no longer be available for use in GCP. However, KeyControl will keep a copy of the deleted cloud key, which can be reloaded back to GCP for use.

1. Sign in to the KeyControl Vault URL bookmark from Create a KeyControl Cloud Key Management Vault.
2. Select the **CLOUDKEYS** icon on the toolbar.
3. Select the **CloudKeys** tab.
4. In the **Key Set** menu, select the **Key Set** created in Create a KeySet for GCP.
5. In the **Key Ring** menu, select the key ring created in Create a GCP key ring.
6. Select the key to the deleted.
7. Select **Actions** > **Delete CloudKey**.

   The **Delete Cloudkey** dialog appears.
8. Enter the number of days when the cloud key should be permanently deleted.
9. Select **Delete**.

10. Verify the Key status changed in KeyControl to **PENDING DELETE**.
11. Verify the key is now scheduled to be deleted from GCP.

    For example:



5.6. Cancel deletion of a deleted KeyControl key

Follow these steps to cancel the deletion and enable back to GCP the KeyControl key deleted in Delete a cloud key in KeyControl.

1. Sign in to the KeyControl Vault URL bookmark from Create a KeyControl Cloud Key Management Vault.
2. Select the **CLOUDKEYS** icon on the toolbar.
3. Select the **CloudKeys** tab.
4. From the **Key Set** menu, select the **Key Set** created in Create a KeySet for GCP.
5. From the **Key Ring** menu, select the key ring created in Create a GCP key ring.
6. Select the key pending to be deleted.
7. Select **Actions** > **Cancel Deletion**.

   The **Cancel Deletion** dialog appears.

8. Select **Yes, Cancel Deletion**.
9. Verify the status change in KeyControl to **DISABLED**.
10. Now enable the key so it is **Available** in GCP.
11. Select **Actions** > **Enable Key**.
12. Verify the status change in KeyControl to **AVAILABLE**.

5.7. Sign/Verify an input file with a GCP CloudKey

This test case uses `gcloud` to sign a file with a GCP cloudkey and OpenSSL to verify the signing.

To install `gcloud`, we use an Ubuntu server to the installation.

1. Install needed packages.

```
% sudo apt-get install apt-transport-https ca-certificates gnupg
```

2. Add `gcloud` cli distribution.

```
echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] https://packages.cloud.google.com/apt cloud-sdk
main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list
```

3. Acquire the Public Key.

```
$ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring
/usr/share/keyrings/cloud.google.gpg add -
```

4. Install `gcloud`.

```
$ sudo apt-get update && sudo apt-get install google-cloud-cli
```

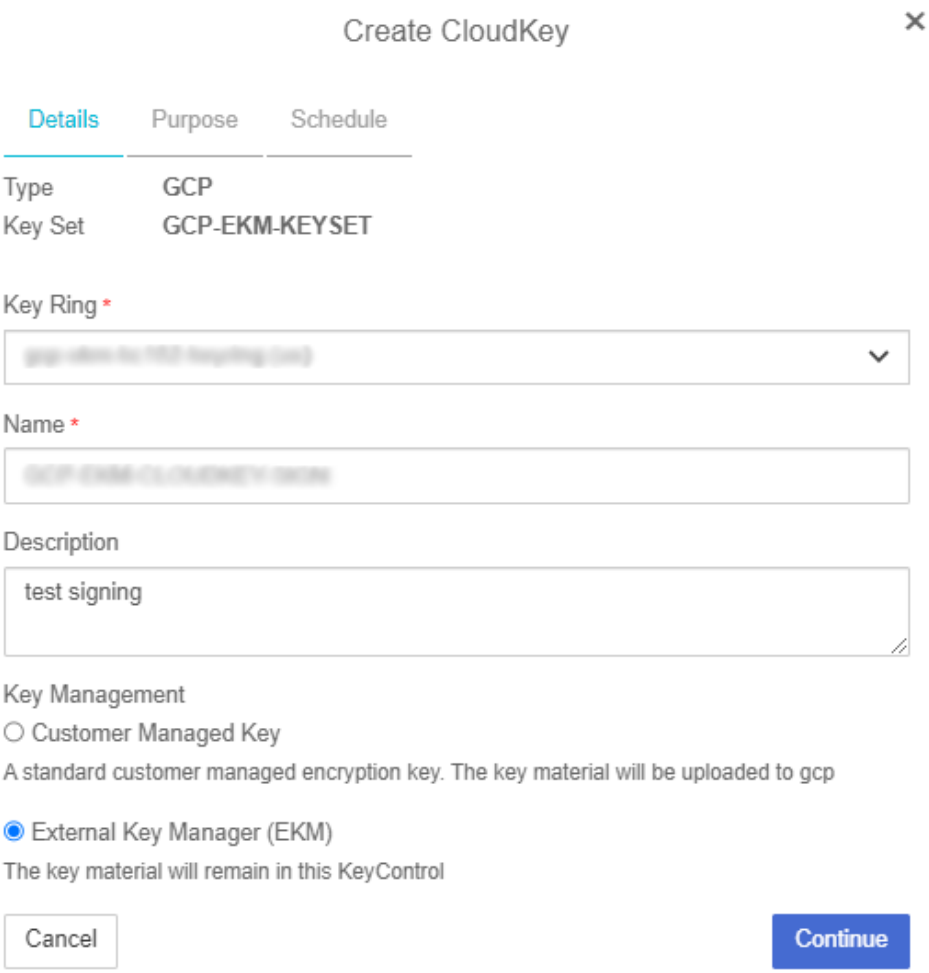
5.8. Create a Cloudkey with purpose of 'Asymmetric Sign'

This procedure is for creating an cloudkey that can be used for signing.

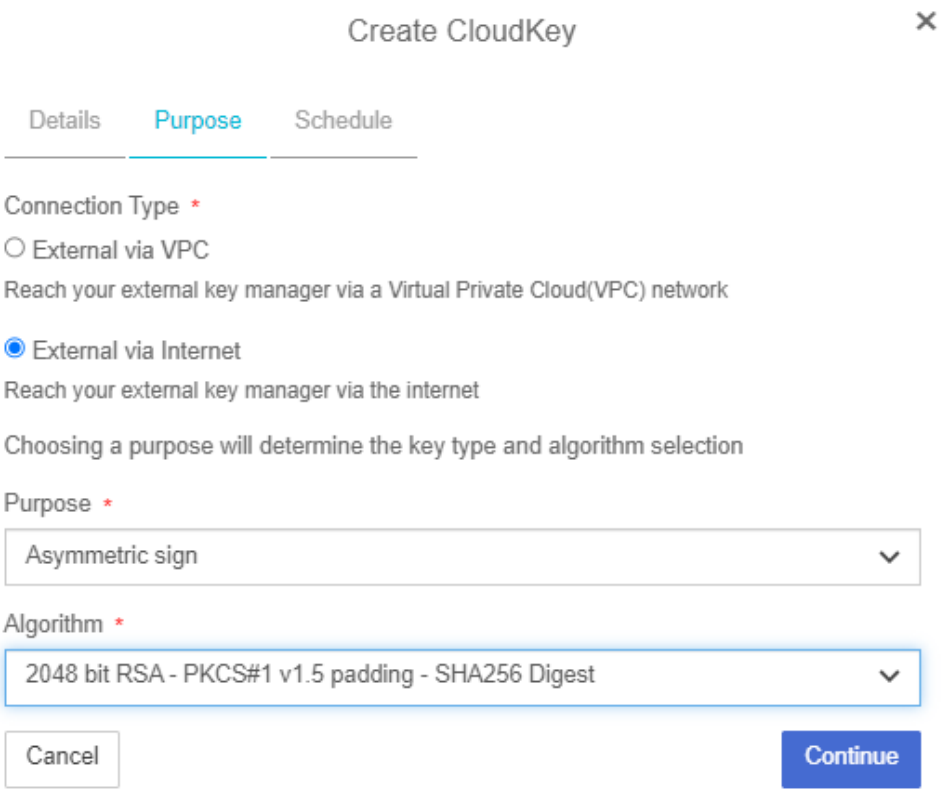
1. Log into the KeyControl Cloud Key Management Vault webGUI using an account with Cloud Admin privileges.
2. In the top menu bar, select **CloudKeys**.
3. Select the **CloudKeys** tab and select the **Key Set** and **Key Ring**.

   If you do not finish the selections on the CloudKeys page, you will need to add them on the Details tab of the Create CloudKey dialog box.

4. Select **Actions** > **Create CloudKey**.
5. On the **Details** tab of the **Create CloudKey** dialog box, enter the following:
   a. **Name**: Enter the name for the CloudKey.
   b. **Description**: Enter the optional description for the CloudKey
   c. **Key Management**: Select **External Key Management (EKM)**.



6. Select **Continue**.
7. On the **Purpose** tab, complete the following:
    a. **Connection Type**: Select how the external key manager will be reached.
    b. **Purpose**: Select **Asymmetric Sign**
    c. **Algorithm**: Select the algorithm **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**



8. Select **Continue**.
9. On the **Schedule** tab, determine the rotation schedule for the CloudKey.
10. Select when the CloudKey should expire. This can be Never, or you can select a specific date.
11. Select **Apply**.

5.9. Use gcloud to sign/verify a file using the cloud key

1. Initialize the `gcloud` cli.

```
$ gcloud init
```

   You need to do this on the console of the machine that has `gcloud` installed. This procedure will attempt to open a browser window for you to sign in to GCP so it can authenticate `gcloud`. Do not ssh and attempt to do this as it will fail to open the browser window if you don't have access to the UI.

2. Set the project

```
$ gcloud config set project htdc-project
```

3. Create a Key for the service account used in the guide.
    a. Open a browser and sign in to the GCP portal: https://console.cloud.google.com.
    b. Select **IAM & Admin** on Google Cloud Menu.
    c. Select **Service Accounts** in the left-hand pane.
    d. Select the service account created in Create a service account in GCP from the list.
    e. Select the **KEYS** tab.
    f. Select **ADD KEY** and then select **Create new key**.
    g. Select **JSON** from the available **Key type** options.
    h. Select **CREATE**.

        A pop-up message appears indicating that the key created was downloaded to your computer.

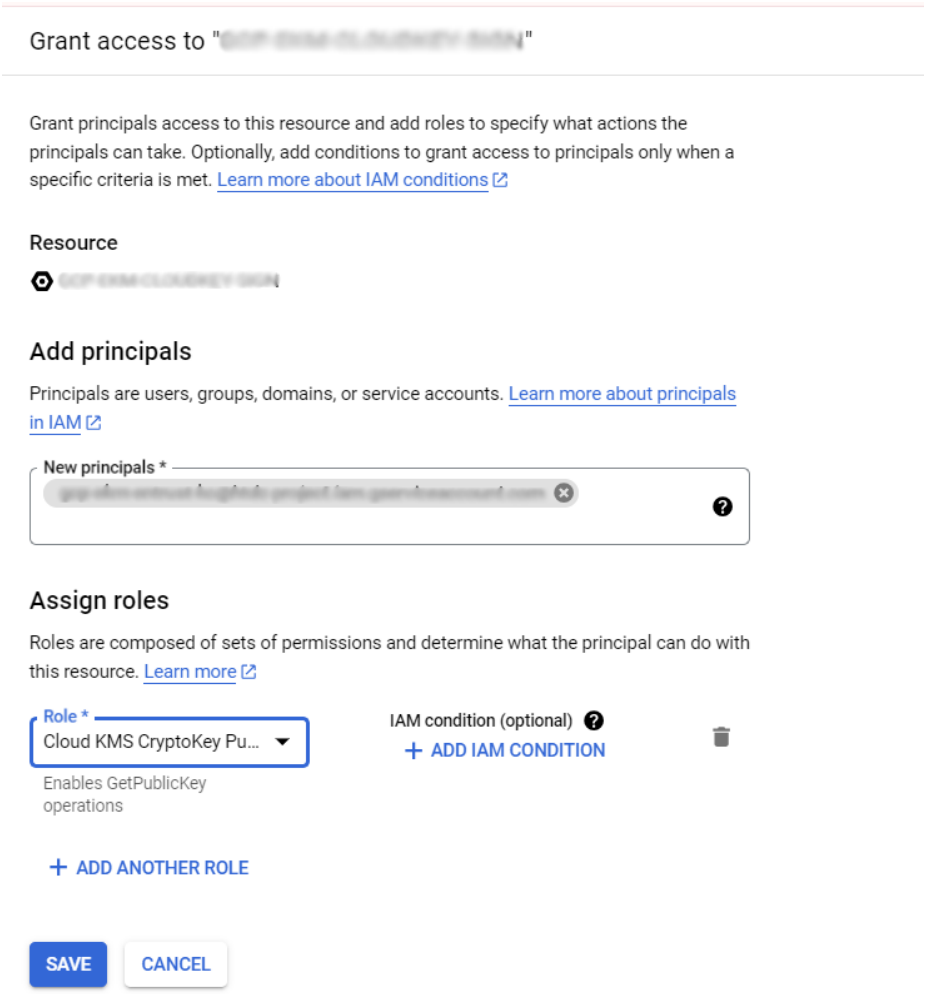
    i. Transfer the json file that was downloaded to the machine you installed `gcloud`.
4. Authorize the Service account

```
$ gcloud auth activate-service-account [Account] --key-file=Key_FILE
```

    Do this in the machine `gcloud` was installed.

```
$ gcloud auth activate-service-account gcp-ekm-entrust-kc@htdc-project.iam.gserviceaccount.com --key
-file=htdc-project-29f147e89a52.json
```

5. Give Permission to the service account to manipulate the key.
    a. In GCP, navigate to **Security** > **Key Management**.
    b. In the **KEY RINGS** tab, select the key ring created earlier.
    c. Select the key that will be used for the signing.
    d. Select **Grant Access**
    e. In the **Grant Access** Pane:
        i. For the **Principal**, enter the service account.
        ii. For the **Role**, select **Cloud KMS Crypto Operator**
        iii. Select **Save**.



6. Download the public key for the cloudkey created earlier.

   Back in the terminal, use `gcloud` to download the public key for the key being used for signing.

   ```
   % gcloud kms keys versions get-public-key 1 --location $location --keyring $keyring --key $key --output
   -file $public_key
   ```

   For example:

   ```
   $ gcloud kms keys versions get-public-key 1 --key GCP-EKM-CLOUDKEY-SIGN --location us --keyring gcp-ekm-
   kc102-keyring --output-file public.key
   ```

7. Create a temp file with some text in it.

   ```
   % vi sign.txt
   ```

   Enter some text in it, for example: `I want to sign this.`

8. Sign the file with `gcloud` cli:

```
$ gcloud kms asymmetric-sign --location $location --keyring $keyring --key $key --version 1 --input-file
inputfile.txt --signature-file $sign_file
```

For example:

```
$ gcloud kms asymmetric-sign --location us --keyring gcp-ekm-kc102-keyring --key GCP-EKM-CLOUDKEY-SIGN
--version 1 --input-file sign.txt --signature-file sign.signed
```

9. Now verify the signed file with download public key using OpenSSL.

```
$ openssl dgst -sha256 -verify $public_key -signature $sign_file inputfile.tx
```

For example:

```
$ openssl dgst -sha256 -verify public.key -signature sign.signed sign.txr
Verified OK
```

If verification is exited with ok , then operation is complete successfully.

Chapter 6. Additional resources and related products