



# CipherTrust Manager Platform



***Google Cloud Bring Your Own Key  
and Bring Your Own Encryption  
tips and techniques***

Document Version 1.3

# Contents

<b>PREFACE .....</b>	<b>3</b>
DOCUMENTATION VERSION HISTORY .....	3
ASSUMPTIONS .....	4
GUIDE TO Thales DOCUMENTATION .....	4
SERVICES UPDATES AND SUPPORT INFORMATION .....	4
<b>GETTING STARTED .....</b>	<b>5</b>
Use Cases .....	5
BYOK Use Case .....	5
BYOK Examples .....	5
BYOK Benefits Summary .....	18
BYOE Use Case .....	19
BYOE Example with Google Big Query .....	19
BYOE Benefits Summary .....	24
<b>Appendix .....</b>	<b>25</b>
Changing from GCP Managed to Customer Managed Keys .....	25
Common Errors .....	25
Converged Thales Vormetric & Gemalto Product Names .....	26
CCKM Sample Reports .....	27
Ports .....	28

# PREFACE

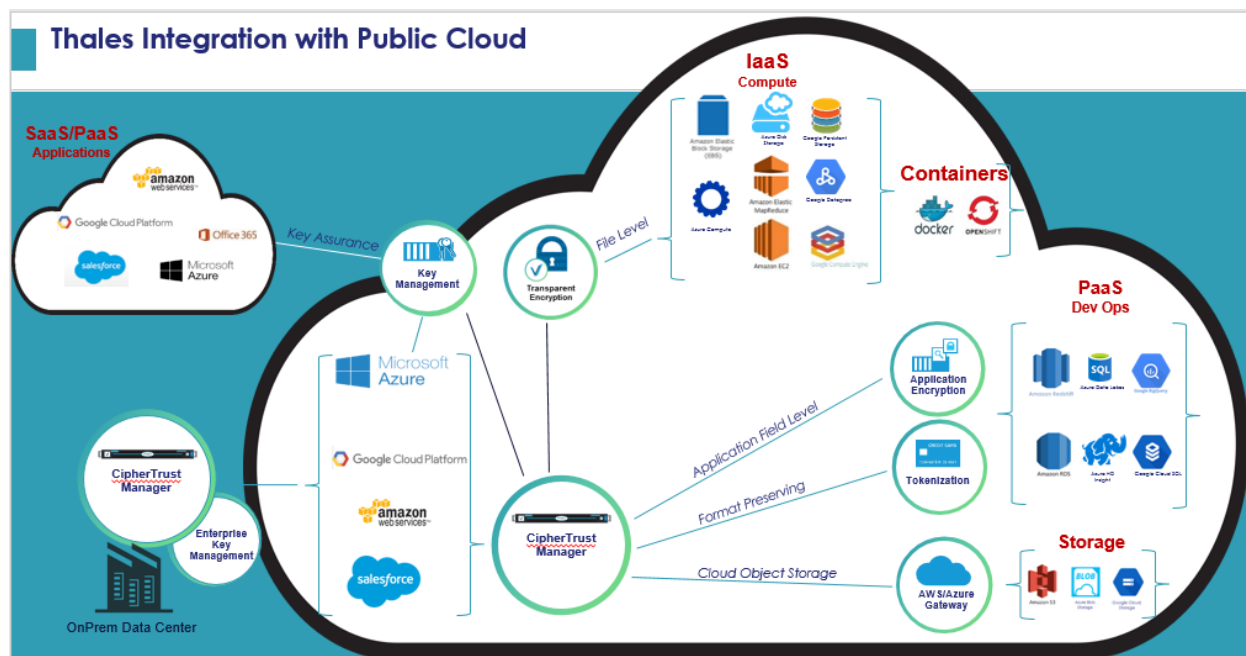
**Note: In Sept of 2020 Thales has rebranded the KeyManager named KeySecure or KeySecure Next Gen to CipherTrust Manager (CM). It combines capabilities from both the legacy Gemalto KeySecure and the Vormetric Data Security Manager products. Any reference in documentation to KeySecure , NextGen or Data Security Manager can be considered to now be the newly branded CipherTrust Manager (CM) product. See following link for more details:**

<https://cpl.thalesgroup.com/encryption/ciphertrust-manager>

GCP BYOE & BYOK tips & techniques, provides examples on how to implement both bringing your own encryption (BYOE) and bringing you own key (BYOK) to the GCP Cloud Provider. Installation steps for the various products are NOT covered in this document since they are covered in the installation guides.

Listed below is a diagram showing the different kinds of cloud offerings by the various cloud providers (IAAS,PAAS,SAAS). BYOK is available for any cloud service that allows for external key import. To find out what services are available please check with each cloud provider. For GCP check the following link: <https://cloud.google.com/security-key-management>

Vormetric BYOE is available for IAAS, PAAS and cloud storage. This document will **not** cover the IAAS for BYOE, since the Vormetric solution using Vormetric Transparent Encryption (VTE), is essentially the same as if the machine were running on premise. *Note: As indicated above the rebranding of Vormetric also applies to VTE. Its new name is CipherTrust Transparent Encryption (CTE).* See appendix for more information on product mappings.



The BYOE examples provided in this document will demonstrate how to implement BYOE for the PAAS Cloud using the CipherTrust Manager Encryption REST API. CipherTrust Manager Developer Suite also includes a SDK called protectApp which can be used as well.

## DOCUMENTATION VERSION HISTORY

Product/Document Version	Date	Changes
V1.3	4/26/21	Modify example to use github helper.

## ASSUMPTIONS

This documentation assumes the reader is familiar with the following Thales products and processes:

- CipherTrust Manager, Gemalto KeySecure or Vormetric Data Security Manager (DSM)
- Key management
- Data encryption
- Familiarity with REST
- Google Cloud

As of November 2020, both CipherTrust Manager and Vormetric Data Security Manager can be used as a key manager for CCKM. The CCKM installation guide describes the detailed steps on how to configure the key source to support CCKM.

## GUIDE TO Thales DOCUMENTATION

Related documents are available to registered users on the Thales Web site at

<https://cpl.thalesgroup.com/> or <https://thalesdocs.com/>

## SERVICES UPDATES AND SUPPORT INFORMATION

The license agreement that you have entered into to acquire the Thales products ("License Agreement") defines software updates and upgrades, support and services, and governs the terms under which they are provided. Any statements made in this guide or collateral documents that conflict with the definitions or terms in the License Agreement, shall be superseded by the definitions and terms of the License Agreement. Any references made to "upgrades" in this guide or collateral documentation can apply either to a software update or upgrade.

# GETTING STARTED

## Use Cases

There are many reasons why customers are looking at both BYOK and BYOE as it relates to the cloud. For example, one independent organization called the Cloud Security Alliance has outlined a best practice for Encryption and Key Management:

Platform and data-appropriate encryption...shall be required and Encryption Keys:  
Shall not be stored in the cloud but  
Shall be maintained by the cloud consumer or trusted key management provider.

Also all cloud service providers state that the security is a shared responsibility. For example Google publishes a document that has a number of suggestions on how to manage the encryption keys at [https://services.google.com/fh/files/misc/gcp\\_pci\\_srm\\_apr\\_2019.pdf](https://services.google.com/fh/files/misc/gcp_pci_srm_apr_2019.pdf)

Keep in mind when implementing BYOK the cloud provider will be implementing the encryption at the disk level which is comparable to full disk encryption. With BYOE, you have complete control over who can access what data and when with your own policies. You also have control over the frequency of the key rotations and re-encryption of the data and audit logs as well. See the following link for more information on BYOE.

<https://es.thalesecurity.com/solutions/use-case/cloud-security/bring-your-own-encryption>

## BYOK Use Case

BYOK for GCP uses the Thales Ciphertrust Cloud Key Manager (CCKM) product. CCKM can be implemented both on premise or in the cloud such as the Azure Marketplace. See following link for more details about CCKM: <https://www.thalesecurity.com/products/key-management/ciphertrust-cloud-key-manager> Support for GCP became available as of CCKM 1.8.0. CCKM leverages the CMEK encryption API's provided by Google.

## BYOK Examples

In order to use BYOK in GCP it is necessary to have a GCP Key Ring. For example, I have both a region specific key ring and a global key ring.

<input type="checkbox"/>	mw-global-key-ring	global	mw-gcp-created-key1, mygcp-sym-key-in-csp2	⋮
<input type="checkbox"/>	mw-us-east1-keyring	us-east1	mw-gcp-created-key2, mygcp-sym-key-in-csp2, m...	▼ ⋮

Step 1. Create a key using CCKM with the Key Sources icon “Add Key” button. Depending on what Thales Key Manager you are using this key will be created on the CipherTrust Manager or the Vormetric

Data Security Manager using a REST API. As an example the CCKM screenshot below shows the results of using a Vormetric Data Security Manager as the Key Manager.

Name	Cloud ID	Key Type	Algorithm	Created Date	Description	Actions
mygcp-asm-key-in-csp	cpcloud.com	GoogleCloud	AES256	10/9/20 5:21 PM		Delete
mygcp-sym-key-in-csp2	cpcloud.com	GoogleCloud	AES256	10/9/20 5:46 PM	mygcp-sym-key-in-csp2	Delete
mygcp-sym-key-in-csp3-bucket	cpcloud.com	GoogleCloud	AES256	10/12/20 11:05 AM	mygcp-sym-key-in-csp3-bucket in csp dsm for gcp buckets	Delete

For this example we created a key called (mygcp-sym-key-in-csp3-bucket)

Step 2. Upload the key from the Thales Key Manager to GCP keyring by selecting the “Keys” icon. Here is an example screenshot:

Key ID	Key Ring	Location	Project ID	Key Purpose	Protection Level	Version	Version State	Key Material Origin	Actions
> mw-gcp-created-key2	mw-us-east1-keyring	South Carolina	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 2	Enabled	KeyRing	Select...
> mw-gcp-created-key1	mw-global-key-ring	global	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 1	Enabled	KeyRing	Select...
> mygcp-sym-key-in-csp2	mw-global-key-ring	global	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 1	Enabled	INTERNAL(mygcp-sym-key-in-csp2)	Select...
> mygcp-sym-key-in-csp2	mw-us-east1-keyring	South Carolina	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 1	Enabled	INTERNAL(mygcp-sym-key-in-csp2)	Select...
> mygcp-sym-key-in-csp3-bu...	mw-us-east1-keyring	South Carolina	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 1	Enabled	INTERNAL(mygcp-sym-key-in-csp3-bucket)	Select...

The key now shows up in the GCP key ring you picked. In this particular example, the keyring is (mw-us-east1-keyring) as show below in the GCP portal.

### Keys for "mw-us-east1-keyring" key ring

A cryptographic key is a resource that is used for encrypting and decrypting data or for producing and verifying digital signatures. To perform operations on data with a key, use the Cloud KMS API. [Learn more](#)

Name	Status	Protection level	Purpose	Next rotation
mw-gcp-created-key2	Available	Software	Symmetric encrypt/decrypt	Apr 4, 2021
mygcp-sym-key-in-csp2	Available	Software	Symmetric encrypt/decrypt	Not scheduled
mygcp-sym-key-in-csp3-bucket	Available	Software	Symmetric encrypt/decrypt	Not scheduled

You can also confirm where keys are from by looking at the “Created from” column below. Any key created by an External Key Manager will have “Import Job”

## Versions for "mygcp-sym-key-in-csp3-bucket" Key

A key version is key material associated with a key at a point in time. A key must have at least one key version to operate on data. Versions are sequentially numbered. You can't view or export a key version. [Learn more](#)

Primary version	Protection level	Purpose	Default algorithm
1 Created on: Oct 12, 2020	Software	Symmetric encrypt/decrypt	Google symmetric key

Filter table

<input type="checkbox"/>	Version	State ?	Algorithm ?	Created on	Created from
<input type="checkbox"/>	1	Enabled & Primary	Google symmetric key	10/12/20, 11:06 AM	Import job

Step 3. In the GCP portal pick any storage bucket in GCP you want to encrypt and select the "Customer-managed key" radio button.

Data is encrypted automatically. Select an encryption key management solution.

- ☐ Google-managed key  
No configuration required
- ☒ Customer-managed key  
Manage via Google Cloud Key Management Service

Select a customer-managed key \*  
mygcp-sym-key-in-csp3-bucket ▼

Step 4. Pick the keyring and the key you want to use.

Step 5. Pick the "mygcp-sym-key-in-csp3-bucket" that was first created in the CipherTrust Manager and then uploaded to the GCP keyring.

Google Cloud Platform

Gemalto-SalesEngineer

Search products and resources

Storage

Browser

Monitoring

Transfer

Transfer for on-premises

Transfer Appliance

Settings

Object details

DOWNLOAD

EDIT METADATA

EDIT PERMISSIONS

DELETE

Buckets > mw-bucket-gcp-cpl > yob1880.txt

Access	Not public
Type	text/plain
Size	24.3 KB
Created	Oct 12, 2020, 11:08:00 AM
Last modified	Oct 12, 2020, 11:08:00 AM
Hold status	None
Retention policy	None
Encryption type	Customer-managed key
Encryption key	<a href="#">projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp3-bucket/cryptoKeyVersions/1</a>
Custom time	—
Public URL	Not applicable
Authenticated URL	<a href="https://storage.cloud.google.com/mw-bucket-gcp-cpl/yob1880.txt">https://storage.cloud.google.com/mw-bucket-gcp-cpl/yob1880.txt</a>
URI	<a href="gs://mw-bucket-gcp-cpl/yob1880.txt">gs://mw-bucket-gcp-cpl/yob1880.txt</a>

Any GCP service that you decide to protect with the mygcp-sym-key-in-csp3-bucket in the GCP keyring will now wrap any GCP DEK key that is created during this process.

## Test BYOK using GCP bucket.

Now access the file you crated in the above. In my example, I used my browser to download the data in my GCP bucket. *Note some GCP services will process the key disable or delete right away and others will have delays that could be up to an hour.*



rm

Gemalto-SalesEngineer

Search

Object details

DOWNLOAD
EDIT METADATA
EDIT PER

Buckets > mw-bucket-gcp-cpl > yob1880.txt

Access	Not public
Type	text/plain
Size	24.3 KB
Created	Oct 12, 2020, 11:08:00 AM
Last modified	Oct 12, 2020, 11:08:00 AM
Hold status	None
Retention policy	None
Encryption type	Customer-managed key
Encryption key	<a href="#">projects/gemalto-salesengineer/locations/</a>
Custom time	—
Public URL ?	Not applicable
Authenticated URL ?	<a href="https://storage.cloud.google.com/mw-buck">https://storage.cloud.google.com/mw-buck</a>
URI ?	<a href="gs://mw-bucket-gcp-cpl/yob1880.txt">gs://mw-bucket-gcp-cpl/yob1880.txt</a>

<https://00f74ba44b>

← → ↻ 🔒

Mary, F, 7065  
Anna, F, 2604  
Emma, F, 2003  
Elizabeth, F, 1939  
Minnie, F, 1746  
Margaret, F, 1578  
Ida, F, 1472  
Alice, F, 1414  
Bertha, F, 1320  
Sarah, F, 1288  
Annie, F, 1258  
Clara, F, 1226  
Ella, F, 1156  
Florence, F, 1063  
Cora, F, 1045  
Martha, F, 1040  
Laura, F, 1012

Now suppose someone accidentally disabled the key in GCP. The user will get the following message in GCP.

## Disable all mygcp-sym-key-in-csp3-bucket versions

Disabled versions can't operate on data. After a key version is disabled, there is a delay of up to a few hours during which it can still operate on data. You can re-enable this key version later. [Learn more](#)

Now shows up as disabled in GCP.

## Keys for "mw-us-east1-keyring" key ring

A cryptographic key is a resource that is used for encrypting and decrypting data or for producing and verifying digital signatures. To perform operations on data with a key, use the Cloud KMS API. [Learn more](#)

Filter table <span>?</span>						
<input type="checkbox"/>	Name <span>↑</span>	Status <span>?</span>	Protection level <span>?</span>	Purpose <span>?</span>	Next rotation <span>?</span>	
<input type="checkbox"/>	mw-gcp-created-key2	Available	Software	Symmetric encrypt/decrypt	Apr 4, 2021	
<input type="checkbox"/>	mygcp-sym-key-in-csp2	Available	Software	Symmetric encrypt/decrypt	Not scheduled	
<input checked="" type="checkbox"/>	<a href="#">mygcp-sym-key-in-csp3-bucket</a>	Not available	Software	Symmetric encrypt/decrypt	Not scheduled	

1 key selected

Since the disable was done in GCP you have to re-sync with CCKM to get an update.

<input type="checkbox"/> Show EKM <span>Export Keys</span> <span>New Key</span> <span>Synchronize</span>			
Version State	Key Material Origin	Rotation Status	Actions
Enabled	KeyRing		Select... ▼
Enabled	KeyRing		Select... ▼
Enabled	INTERNAL(mygcp-sym-key-in-csp2)		Select... ▼
Enabled	INTERNAL(mygcp-sym-key-in-csp2)		Select... ▼
Enabled	INTERNAL(mygcp-sym-key-in-csp3-bucket)		Select... ▼
Showing 1 to 5 of 5 entries			

After re-sync you can see key is disabled.

▼ mygcp-sym-key-in-csp3-bucket	mw-us-east1-keyring	South Carolina	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 1	Disabled	INTERNAL(mygcp
						1	Disabled	INTERNAL(mygcp
... View all versions								

Now try to access the data in the bucket again. Will get the following message.

The Cloud KMS key is disabled or destroyed.

## Enable in CCKM

Disabled	INTERNAL(mygcp-sym-key-in-csp3-bucket)	Select... ▼
Disabled	INTERNAL(mygcp-sym-key-...	<div><div>Enable</div><div>Schedule Destroy</div></div>

Should now see it enabled in GCP.

### Keys for "mw-us-east1-keyring" key ring

A cryptographic key is a resource that is used for encrypting and decrypting data or for producing and verifying digital signatures. To perform operations on data with a key, use the Cloud KMS API. [Learn more](#)

Filter table						
<input type="checkbox"/>	Name ↑	Status ?	Protection level ?	Purpose ?	Next rotation ?	
<input type="checkbox"/>	mw-gcp-created-key2	✓ Available	Software	Symmetric encrypt/decrypt	Apr 4, 2021	⋮
<input type="checkbox"/>	mygcp-sym-key-in-csp2	✓ Available	Software	Symmetric encrypt/decrypt	Not scheduled	⋮
<input type="checkbox"/>	mygcp-sym-key-in-csp3-bucket	✓ Available	Software	Symmetric encrypt/decrypt	Not scheduled	⋮

No keys selected

And can download again.

Mary,F,7065  
Anna,F,2604  
Emma,F,2003  
Elizabeth,F,1939  
Minnie,F,1746  
Margaret,F,1578  
Ida,F,1472  
Alice,F,1414  
Bertha,F,1320  
Sarah,F,1288  
Annie,F,1258  
Clara,F,1226  
Ella,F,1156  
Florence,F,1063  
Cora,F,1045  
Martha,F,1040  
Laura,F,1012  
Nellie,F,995

## Test BYOK using GCP Java Key Samples.

Google provides a number of java code samples that demonstrate the usage of KMS keys. Listed below is a method called “encrypt” from the Google Java SDK sample CryptFile.java example. This particular example took GCP KMS about 1 hour before the key was disabled.

```
public static byte[] encrypt(String projectId, String locationId,
String keyRingId, String cryptoKeyId,
byte[] plaintext) throws IOException
{
System.out.println("in encrypt with project " + projectId + "
location " + locationId + " keyring " + keyRingId
+ "cryptokyeid " + cryptoKeyId);
// Create the KeyManagementServiceClient using try-with-resources
to manage
// client cleanup.
try (KeyManagementServiceClient client =
KeyManagementServiceClient.create()) {

// The resource name of the cryptoKey
String resourceName = CryptoKeyName.format(projectId, locationId,
keyRingId, cryptoKeyId);

// Encrypt the plaintext with Cloud KMS.
EncryptResponse response = client.encrypt(resourceName,
ByteString.copyFrom(plaintext));
byte[] arr = response.getCiphertext().toByteArray();
String base64ciphertext = Base64.getEncoder().encodeToString(arr);
System.out.println("ciphertext in bas64= " + base64ciphertext);
// String base64String = Convert.ToBase64String(arr);
System.out.println("ciphertext binary " +
response.getCiphertext().toByteArray());
// Extract the ciphertext from the response.
return response.getCiphertext().toByteArray();
}

}
```

With the key disabled you can expect to see an error like below.

```
in encrypt with project durable-rhythm-257620 loation global
keyring testcryptokyeid quickstart
Exception in thread "main"
com.google.api.gax.rpc.FailedPreconditionException:
io.grpc.StatusRuntimeException: FAILED_PRECONDITION:
projects/durable-rhythm-
257620/locations/global/keyRings/test/cryptoKeys/quickstart/cryptok
```




```

eyVersions/1 is not enabled, current state is: DISABLED.
at
com.google.api.gax.rpc.ApiExceptionFactory.createException(ApiExcep
tionFactory.java:59)
com.google.cloud.kms.v1.KeyManagementServiceClient.encrypt(KeyManag
ementServiceClient.java:1588)
at
com.google.cloud.kms.v1.KeyManagementServiceClient.encrypt(KeyManag
ementServiceClient.java:1560)
at com.example.CryptFile.encrypt(CryptFile.java:51)
at
com.example.CryptFileCommands$EncryptCommand.run(CryptFileCommands.
java:59)
at com.example.CryptFile.main(CryptFile.java:105)
Caused by: io.grpc.StatusRuntimeException: FAILED_PRECONDITION:
projects/durable-rhythm-
257620/locations/global/keyRings/test/cryptoKeys/quickstart/cryptoK
eyVersions/1 is not enabled, current state is: DISABLED.
at io.grpc.Status.asRuntimeException(Status.java:530)
... 23 more

```

## Test BYOK using GCP Big Query.

For this example I created a new key in the same keyring called mygcp-sym-key-in-csp2. Here is my Google BigQuery dataset properties showing the customer managed key.

Description 	Labels 
None	None
Dataset info 	
Dataset ID	gemalto-salesengineer:mw_dataset_demo
Created	Oct 12, 2020, 10:22:14 AM
Default table expiration	Never
Last modified	Oct 12, 2020, 10:22:14 AM
Data location	us-east1
Default Customer-managed key	projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2

Although not tested Keys can also be associated at a query granularity as well.

**Destination**

☒ Save query results in a temporary table  
☐ Set a destination table for query results

**Project name**  
 Gemalto-SalesEngineer

**Dataset name**  
 babynames

**Table name**  
 Letters, numbers, underscores, and template system characters allowed

**Destination table write preference**  
☐ Write if empty  
☐ Append to table  
☐ Overwrite table

**Results size** ?  
☐ Allow large results (no size limit)

**Resource management**

**Job priority** ?  
☒ Interactive  
☐ Batch

**Cache preference** ?  
☐ Use cached results

**Additional settings**

**SQL dialect** ?  
☒ Standard  
☐ Legacy

**Processing location** ?  
 Auto-select

**Advanced options** ^

**Encryption**  
 Data is encrypted automatically. Select an encryption key management solution.

☐ Google-managed key  
 No configuration required  
☒ Customer-managed key  
 Manage via Google Cloud Key Management Service

**Select a customer-managed key**  
 Keys can be configured in your [Cloud KMS settings](#)  
 us-east1 / mw-us-east1-keyring / mygcp-sym-key-in-csp2

Notice the Customer-managed key along with the No Cached results set below.

names\_2015-limit-10

1
SELECT sum(count) as cnt FROM `gemalto-salesengineer.mw\_dataset\_demo.names\_2015`

Customer-managed key set
No cached results

Run
Save query
Save view
Schedule query
More

Query results

SAVE RESULTSEXPLORE DATA

Query complete (1.1 sec elapsed, 258.9 KB processed)

Job information
Results
JSON
Execution details

Row	cnt
1	3694784

Now suppose there was an incident where someone’s credentials were compromised and the security officer wanted to ensure no one had access to a particular Big Query Dataset. The Thales CCKM administrator can disable the key in CCKM for a short time so the appropriate security administrator could change the compromised person’s credentials. This can be done in the following screen

▼ mygcp-sym-key-in-csp2	mw-us-east1-keyring	South Carolina	gemalto-salesengineer	Symmetric encrypt/decrypt	Software	Primary: 1	Disabled	INTERNAL(mygcp-sym-key-in-csp2)	Select...
						1	Disabled	INTERNAL(mygcp-sym-key-...	Select...

It will be logged in CCKM logs section of the UI, as to who did what and when.

## Event Details: Update\_GCP\_Key\_Version



**Event: Update\_GCP\_Key\_Version**

**Severity: INFO**

**Message:**

Successfully disabled Google Cloud key version projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2/cryptoKeyVersions/1

**User: mark.warner@cplcloud.com**

**Date & Time: 10/13/20 10:00 AM**

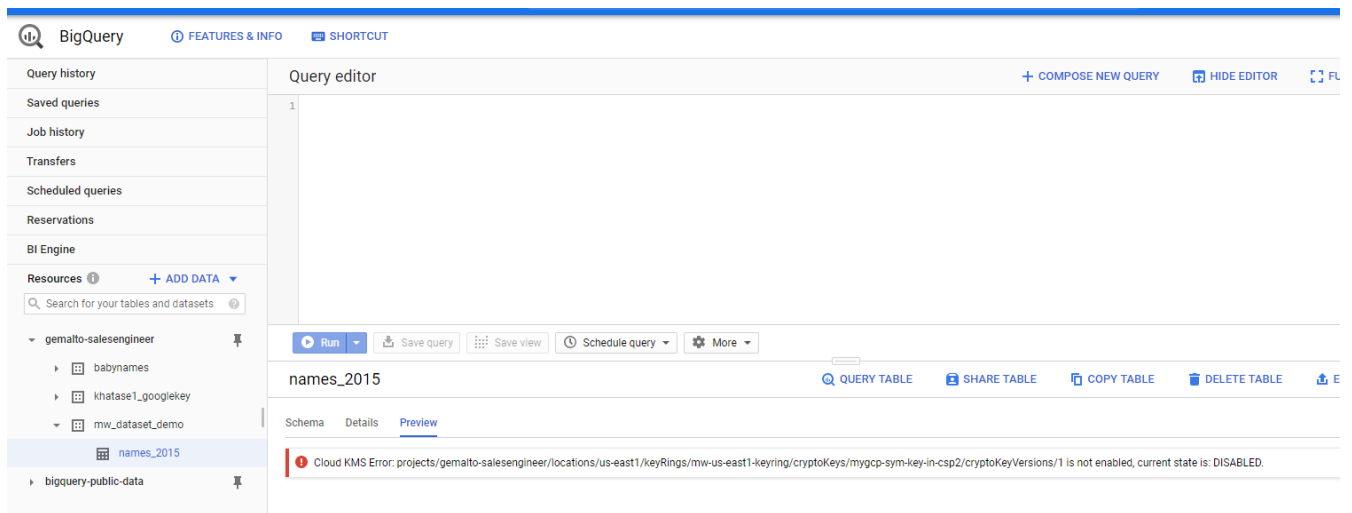
Now shows up as disabled in GCP.

Filter table						
<input type="checkbox"/>	Name ↑	Status ?	Protection level ?	Purpose ?	Next rotation ?	
<input type="checkbox"/>	mw-gcp-created-key2	✓ Available	Software	Symmetric encrypt/decrypt	Apr 4, 2021	⋮
<input type="checkbox"/>	mygcp-sym-key-in-csp2	✗ Not available	Software	Symmetric encrypt/decrypt	Not scheduled	⋮
<input type="checkbox"/>	mygcp-sym-key-in-csp3-bucket	✓ Available	Software	Symmetric encrypt/decrypt	Not scheduled	⋮

Now try to preview the data in Google Big Query. Will get the following message.

Cloud KMS Error: projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2/cryptoKeyVersions/1 is not enabled, current state is: DISABLED.





Enable in CCKM

Should see this in the CCKM logs section of the UI.

**Message:**  
Successfully enabled Google Cloud key version projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2/cryptoKeyVersions/1

Should now see it enabled in GCP

### Keys for "mw-us-east1-keyring" key ring

A cryptographic key is a resource that is used for encrypting and decrypting data or for producing and verifying digital signatures. To perform operations on data with a key, use the Cloud KMS API. [Learn more](#)

Filter table						
<input type="checkbox"/>	Name ↑	Status ?	Protection level ?	Purpose ?	Next rotation ?	
<input type="checkbox"/>	mw-gcp-created-key2	✓ Available	Software	Symmetric encrypt/decrypt	Apr 4, 2021	⋮
<input type="checkbox"/>	mygcp-sym-key-in-csp2	✓ Available	Software	Symmetric encrypt/decrypt	Not scheduled	⋮
<input type="checkbox"/>	mygcp-sym-key-in-csp3-bucket	✓ Available	Software	Symmetric encrypt/decrypt	Not scheduled	⋮

No keys selected

Can run queries again.

names\_2015-limit-10

1 SELECT sum(count) as cnt FROM `gemalto-salesengineer.mw\_dataset\_demo.names\_2015`

Customer-managed key set No cached results

Run

Save query

Save view

Schedule query

More

Query results

SAVE RESULTS

EXPLORE DATA

Query complete (1.1 sec elapsed, 258.9 KB processed)

Job information

Results

JSON

Execution details

Row	cnt
1	3694784

## BYOK Benefits Summary

Listed below is a brief summary of the advantages of using a customer managed key (CMK).

<https://www.thalesecurity.com/products/key-management/ciphertrust-cloud-key-manager>

	GCP Key Ring	CCKM
Create CMEK	Yes, Native or BYOK	Yes, Native , HSM (BYOK) & DSM & CM
Manage CMEKs across multiple Regions	No, you must manually change Regions in the console	Yes, CCKM allows to manage all you CMEKS across regions from a single pane glass
CMEK Versioning	Yes, GCP does allow support key versioning	Yes, CCKM generates a new one and replaces the existing
Delete CMEK	Yes, GCP does not allow to delete the CMEK immediately but enforces a minimum of 24 hours waiting period	Yes, GCP does not allow to delete the CMEK immediately but enforces a minimum of 24 hours waiting period
Restore CMEK	No, once the CMEK is deleted from the Key Ring is unrecoverable	Yes, CCKM reimports the original key material
Rotate CMEK automatically	Yes	Yes
Rotate CMEK based on expiration date	No.	Yes

As mentioned above Vormetric provides BYOE solutions for IAAS, PAAS and cloud storage. This section will cover PAAS and cloud storage scenarios since the IAAS implementation is really the same as the Vormetric on premise solution, which utilizes the VTE agent.

## BYOE Example with Google Big Query

## Implementation steps.

**GCP.**

Create a GCP BigQuery dataset along with the following table.

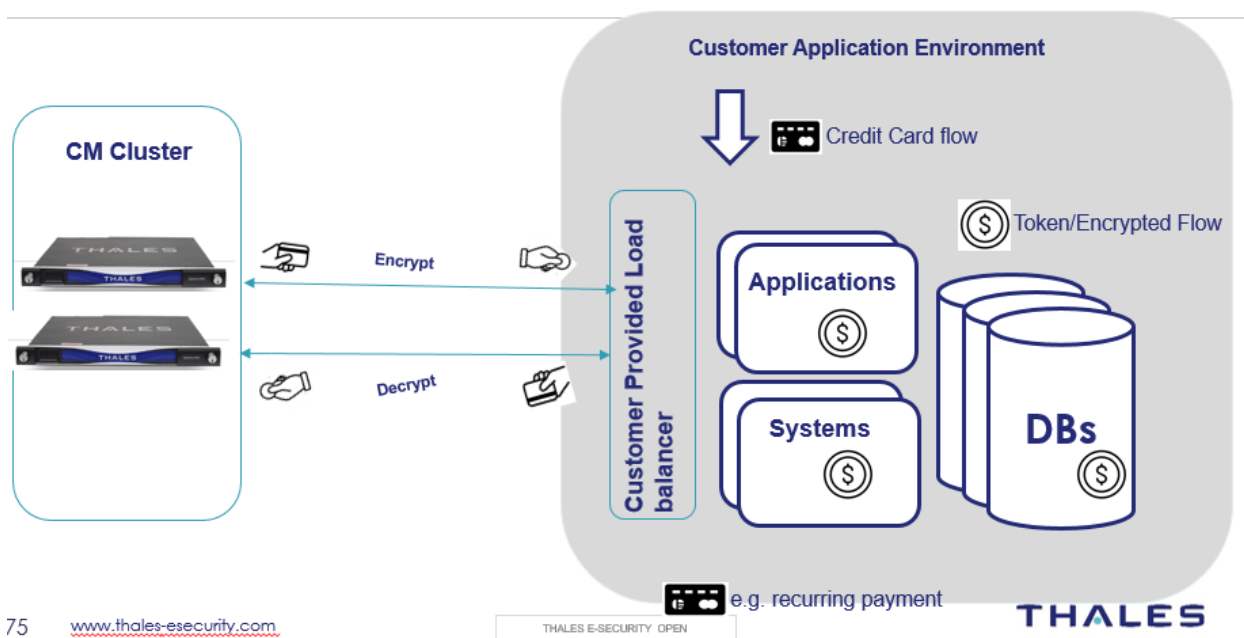
```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

The examples below used an AES 256bit key in CM for encryption along with 15 bytes of random data to be encrypted.

## CipherTrust Manager REST API Example.

Most implementations will have at least two CipherTrust Managers handling requests. The platform operates as a cluster and it is easy to add more nodes to the cluster if needed. As you can see below when using the REST API the customer must provide their own load balancer.

## CipherTrust Manager REST API Deployment



This example uses the CipherTrust Manager REST API to encrypt the data. There are two different modes or algorithms that were used, GCM and Format Preserved Encryption (FPE). These examples use different endpoint URL's and the format of the json payload is different.

Import statements were excluded to keep the document short. Please note this code is for testing only and should not be used for production. As you can see, the only 3<sup>rd</sup> party libraries that were used were for jsonpath to parse json and OkHttpClient to make rest calls and the Simba JDBC drivers for Big Query.

Here is the code to test using standard Big Query JDBC class file and CipherTrust Manager REST API.

## Documentation

- [https://yourcmipaddress/playground\\_v2/api](https://yourcmipaddress/playground_v2/api)

### GCP Big Query JDBC Example.

This sample application uses the BigQuery Simba JDBC drivers to insert data into the Person table using the CipherTrust Manager REST API to protect the Last Name field. The Address field in this example is used store the original data just for comparison purposes. The City field is also used to store the action. Note: This example users a helper class located at:

[https://github.com/thalescpl-io/CipherTrust\\_Application\\_Protection/tree/master/rest/src/main/java/com/thales/cm/rest/cmhelper](https://github.com/thalescpl-io/CipherTrust_Application_Protection/tree/master/rest/src/main/java/com/thales/cm/rest/cmhelper)

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Calendar;
import java.util.Date;
import com.thales.cm.rest.helper.CipherTrustManagerHelper;

public class GCPBigQueryCTMRestApi {

    CipherTrustManagerHelper ctmh = null;

    public static void main(String[] args) throws Exception {

        if (args.length != 8) {
            System.err
                .println("Usage: java AWSMySQLRDSCTMRestApi userid password
keyname numberofrecords batchsize mode operation ctmip ");
            System.exit(-1);
        }

        int numberofrecords = Integer.parseInt(args[3]);
        int batchsize = Integer.parseInt(args[4]);
        String mode = args[5];
        String operation = args[6];

        GCPBigQueryCTMRestApi2 awsresrest = new GCPBigQueryCTMRestApi();

        awsresrest.ctmh = new CipherTrustManagerHelper();
```

```

awsresrest.ctmh.username = args[0];
awsresrest.ctmh.password = args[1];
awsresrest.ctmh.cmipaddress = args[7];

try {
    String tkn = awsresrest.ctmh.getToken();

    awsresrest.ctmh.key = args[2];

}
catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

Calendar calendar = Calendar.getInstance();

// Get start time (this needs to be a global variable).
Date startDate = calendar.getTime();

Connection connection = ConnectionObject.getConnection();

if (mode.equalsIgnoreCase("fpe")) {
    if (operation.equalsIgnoreCase("both")) {
        awsresrest.fpeencrypt( connection, mode, numberofrecords, batchsize);
        awsresrest.fpedecryptdata( connection, mode);
    } else
        awsresrest.fpeencrypt( connection, mode, numberofrecords, batchsize);
} else {
    System.out.println("other encryption modes available");
}

if (connection != null)
    connection.close();

Calendar calendar2 = Calendar.getInstance();

// Get start time (this needs to be a global variable).
Date endDate = calendar2.getTime();
long sumDate = endDate.getTime() - startDate.getTime();
System.out.println("Total time " + sumDate);
}

void fpedecryptdata( Connection connection, String action)
    throws Exception {

    Statement stmt = null;
    try {
        stmt = connection.createStatement();
        String results;

        String sql = "SELECT PersonID, LastName, FirstName, Address, City FROM Persons";
        ResultSet rs = stmt.executeQuery(sql);

        while (rs.next()) {
            // Retrieve by column name

            int id = rs.getInt("PersonID");
            String last = rs.getString("LastName");
            String first = rs.getString("FirstName");
            String addr = rs.getString("Address");
            String city = rs.getString("City");
            System.out.println(", last: " + last);
            // System.out.println("data: " + results);
            System.out.println("ID: " + id);

```

```

        results = this.ctmh.cmRESTProtect( "fpe", last, "decrypt");

        System.out.println("Original Data " + results);
        System.out.println(", First: " + first);
        System.out.println(", addr: " + addr);
    }
    rs.close();

} catch (SQLException se) {
    // Handle errors for JDBC
    se.printStackTrace();
} catch (Exception e) {
    // Handle errors for Class.forName
    e.printStackTrace();
} finally {
    // finally block used to close resources
    try {
        if (stmt != null)
            connection.close();
    } catch (SQLException se) {
    } // do nothing
    try {
        if (connection != null)
            connection.close();
    } catch (SQLException se) {
        se.printStackTrace();
    } // end finally try
} // end try
System.out.println("Goodbye!");
}

void fpeencrypt(Connection connection, String action, int nbrofrecords,
    int batchqty) throws Exception {

    String SQL = "insert into thalesbyoedemo.Persons values (?, ?, ?, ?, ?)";

    int batchSize = batchqty;

    int count = 0;
    int[] result;
    int size = nbrofrecords;
    //simba driver comment out.
    //connection.setAutoCommit(false);
    PreparedStatement pstmt = connection.prepareStatement(SQL);
    String results = null;
    String sensitive = null;

    for (int i = 1; i <= size; i++) {

        sensitive = randomNumeric(15);

        results = this.ctmh.cmRESTProtect( "fpe", sensitive, "encrypt");

        pstmt.setInt(1, i);
        pstmt.setString(2, results);
        pstmt.setString(3, "FirstName");
        pstmt.setString(4, sensitive + " Addr");
        pstmt.setString(5, action);
        pstmt.addBatch();

        count++;

        if (count % batchSize == 0) {
            System.out.println("executeBatch the batch");
            result = pstmt.executeBatch();
            System.out.println("Number of rows inserted: " + result.length);
            //take out for simba
            //connection.commit();
        }
    }
}

```

```

    }

    if (pstmt != null)
        pstmt.close();
    // if(connection!=null)
    // connection.close();

}

// private static final String ALPHA_NUMERIC_STRING =
// "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
private static final String ALPHA_NUMERIC_STRING = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

public static String randomAlphaNumeric(int count) {
    StringBuilder builder = new StringBuilder();
    while (count-- != 0) {
        int character = (int) (Math.random() * ALPHA_NUMERIC_STRING.length());
        builder.append(ALPHA_NUMERIC_STRING.charAt(character));
    }
    return builder.toString();
}

private static final String NUMERIC_STRING = "0123456789";

public static String randomNumeric(int count) {
    StringBuilder builder = new StringBuilder();
    while (count-- != 0) {
        int character = (int) (Math.random() * NUMERIC_STRING.length());
        builder.append(NUMERIC_STRING.charAt(character));
    }
    return builder.toString();
}
}

```

### Example Output

```

Number of rows inserted: 5
Last Name encrypted: FvnrjZiv3JQLHdo1CU0
Decrypted Data: 036065688582665
, First Name: FirstName
, Original Data: 036065688582665 Addr
Last Name encrypted: FfnhqjZgvHldIhtj1iM2
Decrypted Data: 338467575949407
, First Name: FirstName
, Original Data: 338467575949407 Addr
Last Name encrypted: EfPrqzdiuX9RIHdilis0
Decrypted Data: 792575059988485
, First Name: FirstName
, Original Data: 792575059988485 Addr
Last Name encrypted: FPrqjliv3xfKH9u0SQ4
Decrypted Data: 209495667104379
, First Name: FirstName
, Original Data: 209495667104379 Addr

```

Can also use any JDBC or ODBC query tool to access the data, which will always be encrypted. In this case the only method to view unencrypted data is thru an application.

```

386
387 SELECT PersonID, LastName, FirstName, Address, City FROM thalesbyoedemo.Persons;
388
389

```

387:1 [20514] INS

Log 1: Persons [13] x

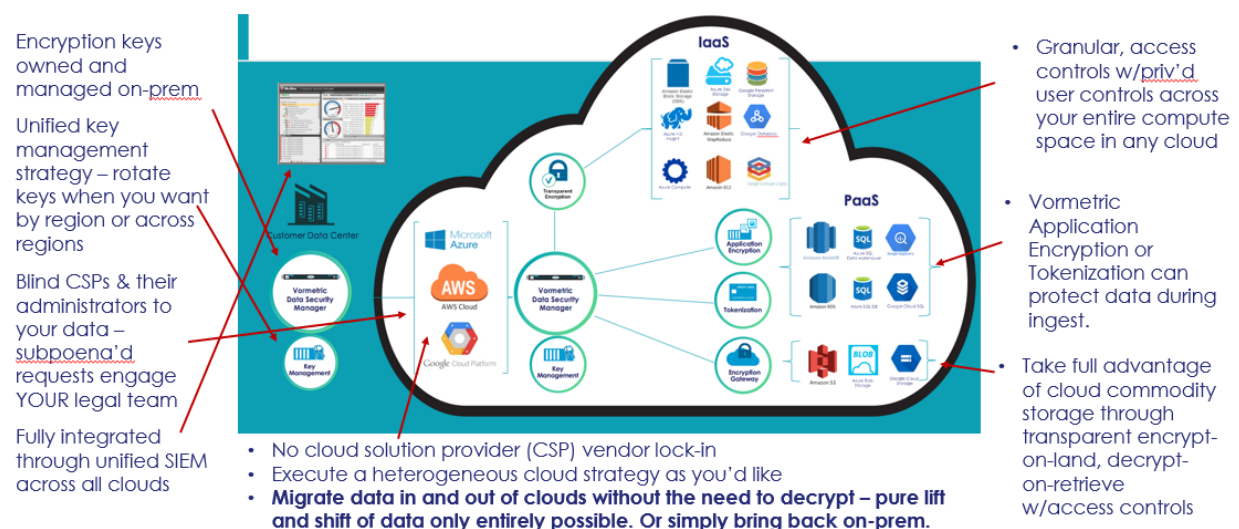


* PersonID	LastName	FirstName	Address	City
1	4 FvnvrjZiv3JQLHdo1CU0	FirstName	036065688582665 Addr	qgWz8kNa7ZFIZPI3o4nsZg==
2	3 FfnhqjZgvH1dIHtj1iM2	FirstName	338467575949407 Addr	qW9tXXjZjgKJnbB7R9nN2g==
3	2 EfPrqzdiuX9RIHdi1is0	FirstName	792575059988485 Addr	3AG09GSCZYecDO0f51/WvQ...
4	2 FPrgqjliv3xfKH9u0SQ4	FirstName	209495667104379 Addr	OQ11puFzVp5x+TJiF84J9Q==
5	3 H/7gqDVusXpdIHZo2yQ0	FirstName	949659805992975 Addr	5pUGiS7IHAVlID78qbMo+g==
6	5 H/nqpjFuvX9aL3pi1CQ5	FirstName	933819452658678 Addr	w0XFLBAp6MXGcL2KLPFBAQ==
7	4 FPjorTRnunJYK39j0CIz	FirstName	221340380209212 Addr	SabaKnUvyY5EISY4urtVEA==
8	2 Ff/pqzRluHxcIXps0Ccw	FirstName	350542164856241 Addr	XkvmW7F7v3l0x56xTFXLQA==
9	5 HvLopzZnvHhbkHlp1CQ5	FirstName	881960523163678 Addr	Udw0rvXsVAypKlYzdtx4Sg==
10	1 E/PvrDFhvH9eIXhi0SUy	FirstName	596216556878363 Addr	JBNqzqQYn8E3wJvQjzZ1bA==
11	3 Ev/hpZuu3xRLHxu0iU5	FirstName	458869269534068 Addr	9xB7sb6WI9OZ45bvypAX8A==
12	1 EvnrrDBhv3heLHlp0yA1	FirstName	432206626563134 Addr	EFSrNmDYMbo9+hcXNsxDsg==
13	1 H/norzVhvHtQK3dv0So1	FirstName	931156518285394 Addr	T+ELMYC4TMQM91C2cIzRtw...

## BYOE Benefits Summary

Listed below is a brief summary of the benefits of using BYOE.

### Vormetric BYOE Cloud Value Proposition

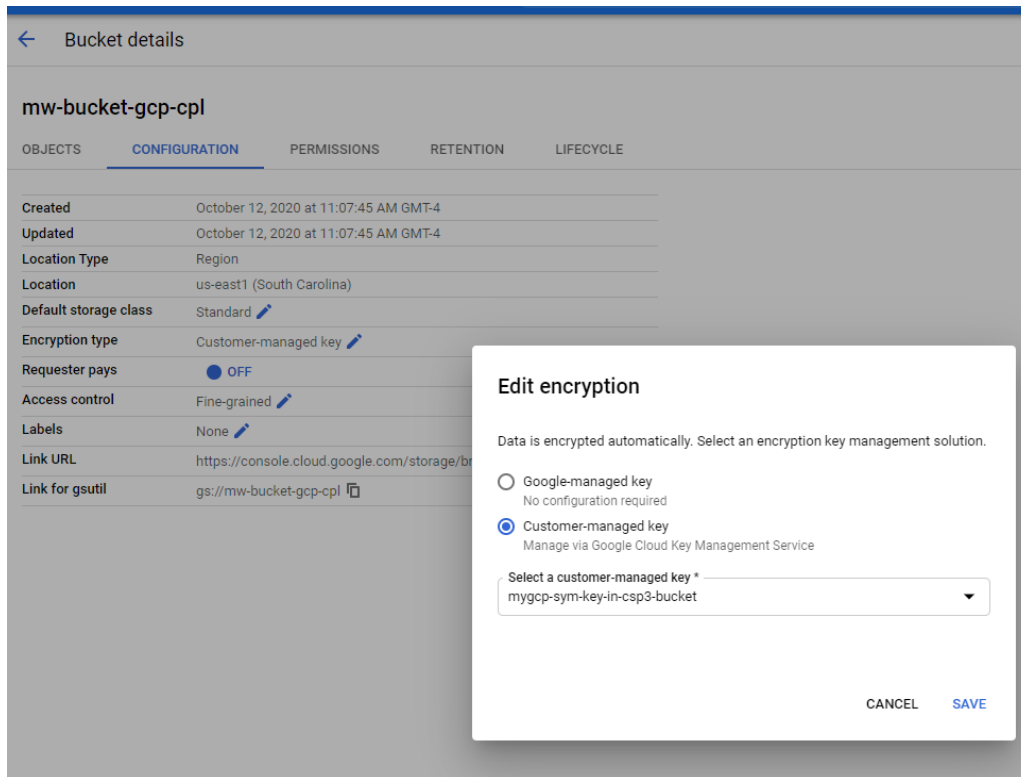




# Appendix

## Changing from GCP Managed to Customer Managed Keys

Some Google Services allow for the changing of keys from Google Managed to Customer Managed Keys. For example, here is where to make the change with Google buckets.



See link below for information on how to change it for Big Query.

<https://cloud.google.com/bigquery/docs/customer-managed-encryption>

## Common Errors

Common error if have not Organization Viewer Access

401 UNAUTHORIZED

This Google Cloud user has no access control on any Google Cloud Key Ring

## Description

None

La

Nor

## Dataset info

Dataset ID	gemalto-salesengineer.babynames
Created	Oct 12, 2020, 10:30:07 AM
Default table expiration	Never
Last modified	Oct 12, 2020, 10:30:07 AM
Data location	us-east1
Default Customer-managed key	projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mw-gcp-created-key2

The screenshot below shows how to set the “Customer Managed Key” for a particular Query.

Destination

☒ Save query results in a temporary table  
☐ Set a destination table for query results

Project name

Gemalto-SalesEngineer

Dataset name


babynames

Table name

Letters, numbers, underscores, and template system characters allowed


Destination table write preference

☐ Write if empty  
☐ Append to table  
☐ Overwrite table


Results size 

☐ Allow large results (no size limit)

Resource management


Job priority 

☒ Interactive  
☐ Batch


Cache preference 

☐ Use cached results


Additional settings

SQL dialect 

☒ Standard  
☐ Legacy

Processing location 

Auto-select

Advanced options 

Encryption

Data is encrypted automatically. Select an encryption key management solution.

☐ Google-managed key  
No configuration required  
☒ Customer-managed key  
Manage via Google Cloud Key Management Service

Select a customer-managed key

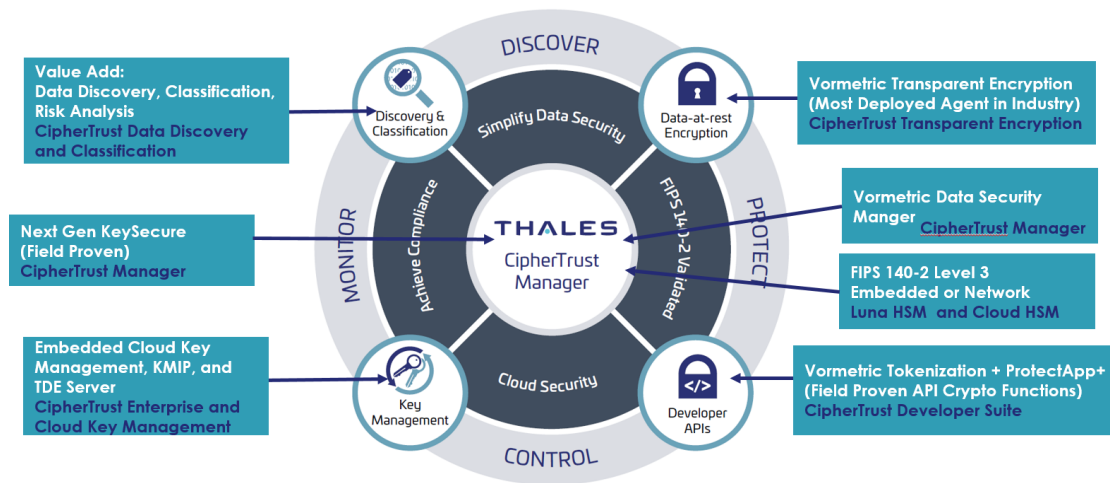
Keys can be configured in your [Cloud KMS settings](#)

us-east1 / mw-us-east1-keyring / mygcp-sym-key-in-csp2

Save

Cancel

# Converged Thales Vormetric & Gemalto Product Names



## CCKM Sample Reports

### Audit Logs.

All Logs (54)				
Event Name	Severity	Date	Event Message	User
			salesengineer/locations/global/keyRings/mw-global-key-ring, projects/gemalto-salesengineer/locations/global/keyRings/sbo_KeyRing, projects/gemalto-salesengineer/locations/global/keyRings/sbo_test, projects/gemalto-salesengineer/locations/us-central1/keyRings/soconnor-cckm-keyring, projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring, projects/gemalto-salesengineer/locations/us-west1/keyRings/test from GoogleCloud	
Synchronize_GCP_Keys	INFO	10/12/20 11:17 AM	Start synchronization	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Upload_GCP_Key	INFO	10/12/20 11:06 AM	Successfully uploaded source key mygcp-sym-key-in-csp3-bucket to Google Cloud key projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp3-bucket	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Add_GCP_Source_Key	INFO	10/12/20 11:05 AM	Successfully created key mygcp-sym-key-in-csp3-bucket in DSM	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Update_GCP_Key_Version	INFO	10/12/20 10:52 AM	Successfully enabled Google Cloud key version projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2/cryptoKeyVersions/1	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Cancel_Delete_GCP_Key_Material	INFO	10/12/20 10:50 AM	Successfully restored Google Cloud key version projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2/cryptoKeyVersions/1	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Login	INFO	10/12/20 10:47 AM	Login to GoogleCloud from 173.76.253.123	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Schedule_Delete_GCP_Key_Material	INFO	10/12/20 10:40 AM	Successfully scheduled Google Cloud key version for destruction projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2/cryptoKeyVersions/1	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Upload_GCP_Key	INFO	10/12/20 10:20 AM	Successfully uploaded source key mygcp-sym-key-in-csp2 to Google Cloud key projects/gemalto-salesengineer/locations/us-east1/keyRings/mw-us-east1-keyring/cryptoKeys/mygcp-sym-key-in-csp2	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Login	INFO	10/12/20 10:19 AM	Login to GoogleCloud from 173.76.253.123	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Upload_GCP_Key	INFO	10/9/20 5:47 PM	Successfully uploaded source key mygcp-sym-key-in-csp2 to Google Cloud key projects/gemalto-salesengineer/locations/global/keyRings/mw-global-key-ring/cryptoKeys/mygcp-sym-key-in-csp2	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Add_GCP_Source_Key	INFO	10/9/20 5:46 PM	Successfully created key mygcp-sym-key-in-csp2 in DSM	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
Change_GCP_Settings	INFO	10/9/20 5:43 PM	Changed Google Cloud alert general settings	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>
ADD_SERVICE_ACCOUNT	INFO	10/9/20 5:42 PM	Added service account 'mw-cckm-demo@gemalto-salesengineer.iam.gserviceaccount.com'	<a href="mailto:mark.warner@colcloud.com">mark.warner@colcloud.com</a>

Reply Forward

## Reconciliation Report

### Combined Key Activity Reconciliation Report

Key Name	Project	Key Ring	Key Activity	Cloud Key Manager Key Activity	Timestamp	Cloud Key Manager Timestamp
mygcp-sym-key-in-csp3-bucket	gemalto-salesengineer	mw-us-east1-keyring	UpdateCryptoKeyPrimaryVersion	Upload_GCP_Key	10/12/20 15:06 UTC	10/12/20 15:06 UTC
AESForLei	gemalto-salesengineer	ApacLei	UpdateCryptoKeyPrimaryVersion		10/12/20 22:35 UTC	
mkcreateswgcpgk1	vp-cckm-dev	mkvpglobalkeyring1	RestoreCryptoKeyVersion		9/17/20 21:51 UTC	
mygcp-sym-key-in-csp2	gemalto-salesengineer	mw-us-east1-keyring	RestoreCryptoKeyVersion	Cancel_Delete_GCP_Key_Material	10/12/20 14:50 UTC	10/12/20 14:50 UTC
mygcp-sym-key-in-csp3-bucket	gemalto-salesengineer	mw-us-east1-keyring	RestoreCryptoKeyVersion	Cancel_Delete_GCP_Key_Material	10/12/20 15:27 UTC	10/12/20 15:27 UTC
MK123	vp-cckm-dev	mkvpglobalkeyring1	RestoreCryptoKeyVersion		10/12/20 21:27 UTC	
MK123A	vp-cckm-dev	mkvpglobalkeyring1	RestoreCryptoKeyVersion		10/12/20 21:28 UTC	
MK123A	vp-cckm-dev	mkvpglobalkeyring1	RestoreCryptoKeyVersion		10/12/20 21:28 UTC	
mygcp-sym-key-in-csp2	gemalto-salesengineer	mw-us-east1-keyring	RestoreCryptoKeyVersion		10/12/20 14:39 UTC	
mkcreateswgcpgk1	vp-cckm-dev	mkvpglobalkeyring1	DestroyCryptoKeyVersion		9/17/20 21:51 UTC	
Show 10 entries						
Showing 31 to 40 of 78 entries						

## Activity Report

### Google Cloud Key Activity Report

Key Name	Project	Key Ring	Key Activity	Region	Origin	Modified By	Modified Time
mw-gcp-created-key1	gemalto-salesengineer	mw-global-key-ring	CreateCryptoKey	global	KeyRing	mark.warner@cplcloud.com	10/06/20 20:53 UTC
mw--gcp-created-key2	gemalto-salesengineer	mw-us-east1-keyring	CreateCryptoKey	us-east1	KeyRing	mark.warner@cplcloud.com	10/06/20 20:56 UTC
TEST_CCKM_LDF	gemalto-salesengineer	cckm_ldf_test	CreateCryptoKey	global	KeyRing	luca.defassi@cplcloud.com	10/09/20 21:22 UTC
mw--gcp-created-key2	gemalto-salesengineer	mw-us-east1-keyring	CreateCryptoKeyVersion	us-east1	KeyRing	mark.warner@cplcloud.com	10/09/20 21:29 UTC
mw--gcp-created-key2	gemalto-salesengineer	mw-us-east1-keyring	UpdateCryptoKeyPrimaryVersion	us-east1	KeyRing	mark.warner@cplcloud.com	10/09/20 21:29 UTC
LeiTESTkey	gemalto-salesengineer	ApacLei	UpdateCryptoKeyVersion	asia-southeast2	KeyRing	lei.zhao@cplcloud.com	10/11/20 23:05 UTC
asd1	gemalto-salesengineer	test	CreateCryptoKey	us-west1	KeyRing	luca.defassi@cplcloud.com	10/12/20 09:00 UTC
asd1	gemalto-salesengineer	test	UpdateCryptoKeyPrimaryVersion	us-west1	KeyRing	luca.defassi@cplcloud.com	10/12/20 09:00 UTC
test-gcp-key	gemalto-salesengineer	Test	CreateCryptoKey	global	KeyRing	luca.defassi@cplcloud.com	10/12/20 10:37 UTC
test-123	gemalto-salesengineer	cckm_ldf_test	CreateCryptoKey	global	KeyRing	luca.defassi@cplcloud.com	10/12/20 12:52 UTC
Show 10 entries							
Showing 1 to 10 of 46 entries							

## User Action Report

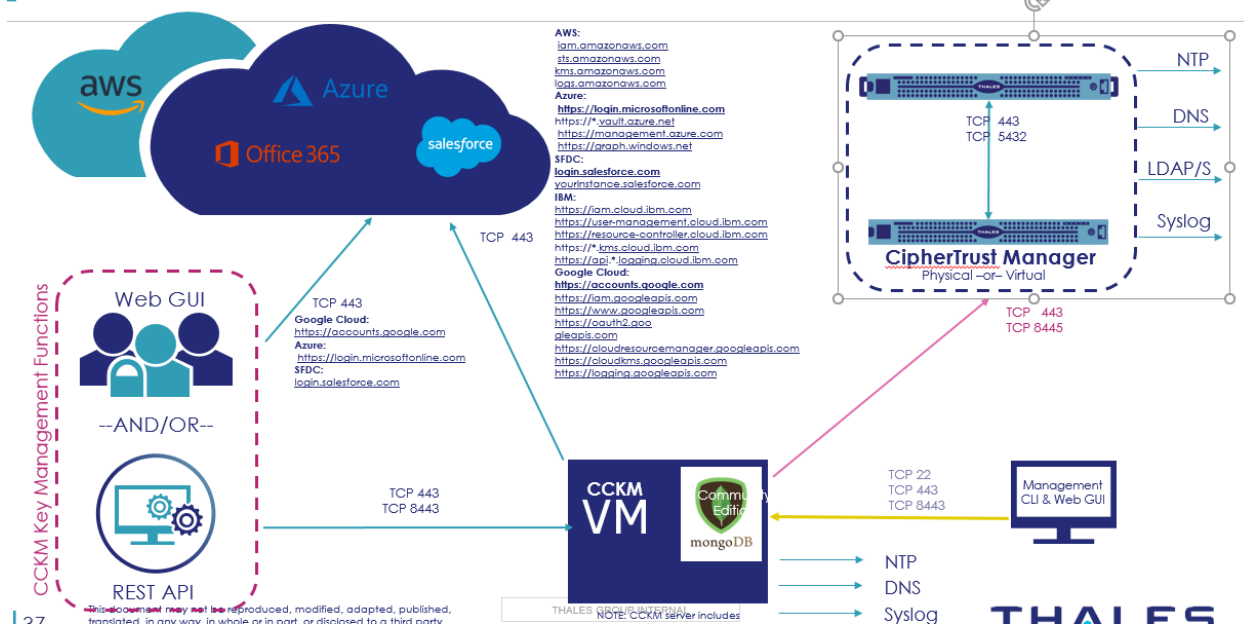
### Cloud Key Manager User Action Report

User Name	Upload Key	Schedule Destroy	Cancel Delete	Rotate Key
mark.warner@cplcloud.com	4	2	2	1
Show 10 entries				
Showing 1 to 1 of 1 entries				

## Ports

As indicated earlier CCKM Enterprise supports both CipherTrust Manager and Vormetric Data Security Manager as a key source. The diagram below shows the ports and endpoints needed for CCKM using the CipherTrust Manager as the key manager.

## CCKM Enterprise POC Deployment CipherTrust Manager



The diagram below shows the ports and endpoints needed for CCKM using the Vormetric Data Security Manager as the key manager.

## CCKM Proof of Concept Deployment Vormetric DSM

