This page discusses customer-managed encryption keys. For other encryption options, see Data Encryption Options (/storage/docs/encryption). For examples of using this feature, see Using Customer-Managed Encryption Keys (/storage/docs/encryption/using-customer-managed-keys).

As an additional layer on top of Google-managed encryption keys (/storage/docs/encryption/default-keys), you can choose to use keys generated by Cloud Key Management Service. Such keys are known as *customer-managed encryption keys*. If you use a customer-managed encryption key, your encryption keys are stored within Cloud KMS. The project that holds your encryption keys can then be independent from the project that contains your buckets, thus allowing for better separation of duties (/kms/docs/separation-of-duties).

When you apply a customer-managed encryption key to an object, Cloud Storage uses the key when encrypting:

- The object's data.
- The object's CRC32C checksum.
- The object's MD5 hash.

Cloud Storage uses standard server-side keys (/storage/docs/encryption/default-keys) to encrypt the remaining metadata (/storage/docs/metadata) for the object, including the object's name. Thus, if you have sufficient permission (/storage/docs/access-control/index), you can perform actions such as reading most metadata, listing objects, and deleting objects even after you've disabled or destroyed the associated customer-managed encryption key.

Because most metadata can be read regardless of encryption method, do not include sensitive information in your o
ucket metadata or names.

Encryption and decryption with customer-managed encryption keys is accomplished using service accounts. Once you give your Cloud Storage service account access (/storage/docs/encryption/using-customer-managed-keys#service-account-access) to an encryption key, that service account encrypts:

- Objects added to a bucket that uses the key as the default key (/storage/docs/encryption/using-customer-managed-keys#add-default-key).
- Specific objects that you indicate should be encrypted with that key (/storage/docs/encryption/using-customer-managed-keys#add-object-key).

When adding or rewriting an object in Cloud Storage, if you have both a default key set on your bucket and a specific key included in your request, Cloud Storage uses the specific key to encrypt the object.

When a requester wants to read an object encrypted with a customer-managed encryption key, they simply access the object as they normally would. During such a request, the service account automatically decrypts the requested object as long as:

- The service account still has permission to decrypt using the key.
- You have not disabled or destroyed the key.

If one of these conditions is not met, the service account does not decrypt the data, and the request fails.

tant: There is a delay between when you disable a key (/kms/docs/enable-disable#disable_an_enabled_key_version) that key no longer can be used to encrypt and decrypt objects. For information about the typical length of the delay a o effectively shorten it, see Cloud KMS Resource Consistency (/kms/docs/consistency#key_versions).

n: If you destroy a key (/kms/docs/destroy-restore) that encrypts existing objects, you will be unable to recover that u continue to be charged for storage of your objects until you delete them. If you destroy a key, use the object metad **msKeyName** to identify objects that used the destroyed key, and delete those objects.

A Cloud KMS key resource has the following format:

Where:

- `[PROJECT_STORING_KEYS]` is the ID of the project associated with the key. For example, `my-pet-project`.
- `[LOCATION]` is the key location (/kms/docs/locations). For example, `US-EAST1`.
- `[KEY_RING_NAME]` is the name of the key ring. For example, `my-key-ring`.
- `[KEY_NAME]` is the name of the key. For example, `my-key`.

The following restrictions apply when using customer-managed encryption keys:

- Cloud SQL exports to Cloud Storage (/sql/docs/mysql/import-export/exporting) and Dataflow (/dataflow/) do not currently support objects encrypted with customer-managed encryption keys.
- You cannot use the JSON API Copy Object method (/storage/docs/json_api/v1/objects/copy) when the source object is encrypted with a customer-managed encryption key or when the destination object would become encrypted by a customer-managed encryption key. Use the Rewrite Object method (/storage/docs/json_api/v1/objects/rewrite) instead.
- You cannot use the JSON API Compose Object method (/storage/docs/json_api/v1/objects/compose) when one or more of the source objects are encrypted with a customer-managed encryption key.
- You cannot encrypt an object with a customer-managed encryption key by updating the object's metadata. Include the key as part of a rewrite of the object instead.
- You must create the Cloud KMS key in the same location as the data you intend to encrypt. For example, if your bucket is located in `US-EAST1`, any Cloud KMS key encrypting objects in that bucket must also be created in `US-EAST1`. For available Cloud KMS locations, see Cloud KMS locations (/kms/docs/locations).
- You cannot specify a customer-managed encryption key as part of a Storage Transfer Service (/storage-transfer/docs/overview) transfer, and any such keys on source objects are not applied to the transferred objects. Set a default customer-managed key on your bucket (/storage/docs/encryption/using-customer-managed-keys#add-default-key) prior to performing the transfer.

In addition to customer-managed encryption, Cloud Storage offers Customer-Supplied Encryption Keys (/storage/docs/encryption/customer-supplied-keys) as a way of controlling your data encryption. You can encrypt different objects in a single bucket with different encryption methods, but note that:

- A single object can only be encrypted by one of these methods at a time.
- If you have a default customer-managed key set for your bucket and specify a customer-supplied key in a request, Cloud Storage uses the customer-supplied key to encrypt the object.
- You can set a default customer-**managed** key on your bucket (/storage/docs/encryption/using-customer-managed-keys#add-default-key), but you cannot set a default customer-**supplied** key on your bucket.