This page describes how to use a Cloud Key Management Service encryption key with Cloud Storage, including getting started with the feature, using default keys on buckets, and adding keys to individual objects. The Cloud KMS encryption key is a *customer-managed encryption key*, which is created by Cloud KMS and managed by you. For details about this feature, including countries where it's available, see Customer-Managed Encryption Keys (/storage/docs/encryption/customer-managed-keys). For other encryption options in Cloud Storage, see Data Encryption Options (/storage/docs/encryption).

Before using this feature in Cloud Storage, you should:

1. Enable the Cloud KMS API for the project that will store your encryption keys.

   Enable the API (https://console.cloud.google.com/flows/enableapi?apiid=cloudkms.googleapis.com)

2. Have sufficient permission on the project that will store your encryption keys:

   - If you own the project that will store your keys, you most likely have the necessary permission.
   - If you plan to create new encryption key rings and keys, you should have `cloudkms.keyRings.create` and `cloudkms.cryptoKey.create` (/kms/docs/reference/permissions-and-roles) permission.
   - Whether you plan to use new or existing key rings and keys, you should have `cloudkms.cryptoKey.setIamPolicy` (/kms/docs/reference/permissions-and-roles) permission for the keys that you will use for encryption.

     This permission allows you to give Cloud Storage service accounts access to Cloud KMS keys.
   - The above permissions are contained in the `roles/cloudkms.admin` role.

     See Using IAM with Cloud KMS (/kms/docs/iam) for instructions on how to get this or other Cloud KMS roles.

3. Have sufficient permission to work with objects in your Cloud Storage bucket:

   - If you own the project that contains the bucket, you most likely have the necessary permission.

- If you use IAM, you should have `storage.objects.create` permission (/storage/docs/access-control/iam#permissions) to write objects to the bucket and `storage.objects.get` permission to read objects from the bucket. See Using IAM Permissions (/storage/docs/access-control/using-iam-permissions#bucket-add) for instructions on how to get a role, such as `roles/storage.objectAdmin` that has these permissions.
- If you use ACLs, you should have bucket-scoped `WRITER` permission (/storage/docs/access-control/lists#permissions) to write objects to the bucket and object-scoped `READER` permission to read objects from the bucket. See Setting ACLs (/storage/docs/access-control/create-manage-lists#set-an-acl) for instructions on how to do this.

4. Have a Cloud KMS key ring (/kms/docs/creating-keys#create_a_key_ring), and have at least one key (/kms/docs/creating-keys#create_a_key) within the key ring.

Note: You must create the Cloud KMS key in the same location as the data you intend to encrypt. For available Cloud KMS locations, see Cloud KMS locations (/kms/docs/locations).

5. **Note:** The following step is not necessary if you are using the gsutil tool.

Get the email address of the service account (/storage/docs/getting-service-account) associated with the project that contains your Cloud Storage bucket.

In order to use customer-managed encryption keys, give your Cloud Storage service account permission to use your Cloud KMS key:

To add or change the Cloud KMS key that is used by default when objects are written to a bucket:

Objects that were written to a bucket prior to adding or changing the default key remain encrypted with the previous
ption method.

To view the Cloud KMS key that is currently set as default for your bucket:

To remove any default Cloud KMS key set on a bucket:

Objects that were written to a bucket prior to removing the default key and which used the default key remain encryp

You can encrypt an individual object with a Cloud KMS key. This is useful if you want to use a different key from the default key set on the bucket, or if you don't have a default key set on the bucket.

To find the name of the Cloud KMS key that was used to encrypt an object:

If the object was written with an encryption method other than Cloud KMS, there is no information associated with t
ata field.

Decrypting an object encrypted with a customer-managed encryption key is performed automatically as long as the relevant service account has access to the key. For more information, see Service accounts with customer-managed encryption keys (/storage/docs/encryption/customer-managed-keys#service-accounts).

- Learn more about customer-managed encryption keys (/storage/docs/encryption/customer-managed-keys).
- Learn more about Cloud Key Management Service (/kms/).
- Learn about customer-supplied encryption keys (/storage/docs/encryption/customer-supplied-keys).