Secure Cloud Storage

Identity and Access Management

IAM allows you to control who has access to the *resources* in your Google Cloud project. Resources include Cloud Storage buckets and objects stored within buckets, as well as other Google Cloud entities such as Compute Engine instances.

The set of access rules you apply to a resource is called an IAM *policy*. An IAM policy applied to your project defines the actions that users can take on all objects or buckets within your project. An IAM policy applied to a single bucket defines the actions that users can take on that specific bucket and objects within it.

Should you use access control lists?

In most cases, Identity and Access Management (IAM) is the recommended method for controlling access to your resources. IAM and ACLs work in tandem to grant access to your buckets and objects: a user only needs permission from either IAM or an ACL to access a bucket or object.

You most likely want to use ACLs if you need to customize access to individual objects within a bucket, since IAM permissions apply to all objects within a bucket. However, you should still use IAM for any access that is common to all objects in a bucket, because this reduces the amount of micro-managing you have to do.

Create and manage access control lists (ACLs)

https://cloud.google.com/storage/docs/access-control/create-manage-lists#set-an-acl

Data encryption options

Cloud Storage always encrypts your data on the server side, before it is written to disk, at no additional charge.

- *Server-side encryption*: encryption that occurs after Cloud Storage receives your data, but before the data is written to disk and stored.
    - *Customer-managed encryption keys*: You can create and manage your encryption keys through Cloud Key Management Service. Customer-managed encryption keys can be stored as software keys, in an HSM cluster, or externally.
    - *Customer-supplied encryption keys*: You can create and manage your own encryption keys. These keys act as an additional encryption layer on top of the standard Cloud Storage encryption.
- *Client-side encryption*: encryption that occurs before data is sent to Cloud Storage. Such data arrives at Cloud Storage already encrypted but also undergoes server-side encryption.

Encrypt Data:

Google-managed encryption keys

Cloud Storage always encrypts your data on the server side, before it is written to disk, at no additional charge. This page discusses the standard encryption that Cloud Storage performs. For other encryption options, see Data Encryption Options.

Cloud Storage manages server-side encryption keys on your behalf using the same hardened key management systems that we use for our own encrypted data, including strict key access controls and auditing. Cloud Storage encrypts user data at rest using AES-256.

Customer-managed encryption keys

If you need more control over key operations than what Google-managed encryption keys allows, you can use *customer-managed encryption keys*. These keys are created and managed using Cloud Key Management Service, and you store the keys as software keys, in an HSM cluster, or externally. You can use customer-managed encryption keys on individual objects, or configure your bucket to use a key by default on all new objects added to a bucket.

Customer-supplied encryption keys

As an additional layer on top of Google-managed encryption keys, you can choose to provide your own AES-256 key, encoded in standard Base64. This key is known as a *customer-supplied encryption key*. If you provide a customer-supplied encryption key, Cloud Storage does not permanently store your key on Google's servers or otherwise manage your key.

Instead, you provide your key for each Cloud Storage operation, and your key is purged from Google's servers after the operation is complete. Cloud Storage stores only a cryptographic hash of the key so that future requests can be validated against the hash. Your key cannot be recovered from this hash, and the hash cannot be used to decrypt your data.

Client-side encryption keys

When you perform client-side encryption, you must create and manage your own encryption keys, and you must use your own tools to encrypt data prior to sending it to Cloud Storage. Data that you encrypt on the client side arrives at Cloud Storage in an encrypted state, and Cloud Storage has no knowledge of the keys you used to encrypt the data.