This page discusses hash-based message authentication code (HMAC) keys, which you can use to authenticate requests to Cloud Storage. For a guide to creating and managing service account HMAC keys, see Managing HMAC keys for service accounts (/storage/docs/authentication/managing-hmackeys).

An HMAC key is a type of *credential* and can be associated with a service account or a user account in Cloud Storage. You use an HMAC key to create *signatures* which are then included in requests to Cloud Storage. Signatures show that a given request is authorized by the user or service account.

HMAC keys have two primary pieces, an *access ID* and a *secret*.

- **Access ID**: A 24 character alphanumeric string, which is linked to your Google account. An example of an access ID:

  `GOOGTS7C7FUP3AIRVJTE2BCD`

- **Secret**: A 40 character Base-64 encoded string that is linked to a specific access ID. A secret is a preshared key that only you and Cloud Storage know. You use your secret to create signatures as part of the authentication process. An example of a secret:

  `bGoa+V7g/yqDXvKRqq+JTFn4uQZbPiQJo4pf9RzJ`

Both the access ID and secret uniquely identify an HMAC key, but the secret is much more sensitive information, because it's used to create signatures.

HMAC keys are useful when:

- You want to move data between other cloud storage providers and Cloud Storage, because HMAC keys allow you to reuse your existing code (/storage/docs/migrating#migration-simple) to access Cloud Storage.

When you create an HMAC key for a service account, you are given the secret for the key once. You must securely store the secret, along with the associated *access ID*. If you lose the secret, it cannot be retrieved by you or Google, and you must create a new HMAC key for the service account to continue authenticating requests.

When you create an HMAC key for a user account, you can view the key's secret from the Google Cloud Console. To do so, you must be logged into the Cloud Console with the user account. Secrets associated with the user account are found in the Cloud Storage **Settings** menu, in the **Interoperability** tab.

- Do not share your HMAC key secret. You should treat HMAC key secrets as you would any set of access credentials.
- As a security best practice, you should regularly change your keys as part of a key rotation.
- If you think someone else is using your HMAC keys, you should immediately delete the affected HMAC keys and create new ones.
- When changing HMAC keys, you should update your code with the new HMAC keys before you delete the old keys. When you delete HMAC keys, they become immediately invalid, and they are not recoverable.

- HMAC keys can only be used to make requests to the XML API, not the JSON API.
- You can have a maximum of 5 HMAC keys per service account. Deleted keys do not count towards this limit.

Generally, associating HMAC keys with service accounts are a better option than doing so with user accounts, particularly for production workloads:

- Service accounts allow for better administrative oversight, and they eliminate the security and privacy implications of accounts held by individual users.
- Service accounts reduce the risk of service outages associated with relying on user accounts, such as when a user account is disabled because the user leaves the project or company.

If you currently use HMAC keys with user accounts but want to migrate to service accounts, keep the following in mind:

- Your project must have a service account (/iam/docs/creating-managing-service-accounts#creating_a_service_account) and have an HMAC key (/storage/docs/authentication/managing-hmackeys#create) associated with it.
- The service account must be granted the required permissions (/storage/docs/access-control/iam-permissions) to perform actions in Cloud Storage.

  Broad permission to work with objects is contained in the `Storage Object Admin` role, but you may want to have separate service accounts for performing different actions. For example, you may want one service account for reading, which would have the `Storage Object Viewer` role and a second service account for writing, which would have the `Storage Object Creator` role.
- You should test to make certain the service account behaves as expected before pushing any update out to production.
- After your production work transitions to service account HMAC keys, you should check the following Stackdriver metric (/monitoring/api/metrics_gcp#gcp-storage) to verify that the HMAC keys associated with the user account are no longer in use:

Metric	Description
`storage.googleapis.com/authn/authentication_count`	The number of times HMAC keys have been used to authenticate requests.

  You can set the following labels to track user account keys that are still in use during the migration progress:

  - `access_id`: identifies which access ID made the request. You can also use `access_id` during a key rotation to watch traffic move from one key to another.
  - `authentication_method`: identifies if keys are user account or service account keys.

- Once you've verified the user account HMAC keys are no longer used, you should delete those HMAC keys. Doing so reduces the risk of inappropriate data access.
- If the user account is no longer used to access Cloud Storage resources, revoke any access to Cloud Storage that it has.

- Create HMAC keys for your service accounts (/storage/docs/authentication/managing-hmackeys#create).
- Use an HMAC key in an authenticated request (/storage/docs/migrating#authentication).