

[Security & Identity Products](https://cloud.google.com/products/security/) (<https://cloud.google.com/products/security/>)

[Cloud IAM](https://cloud.google.com/iam/) (<https://cloud.google.com/iam/>)

[Documentation](https://cloud.google.com/iam/docs/) (<https://cloud.google.com/iam/docs/>) [Guides](#)

# Creating and managing service account keys

This page explains how to create and manage service account keys using the Google Cloud Console, the [gc1oud command-line tool](https://cloud.google.com/sdk/gcloud/) (<https://cloud.google.com/sdk/gcloud/>), the [Cloud Identity and Access Management API](https://cloud.google.com/iam/reference/rest/) (<https://cloud.google.com/iam/reference/rest/>), or one of the [Google Cloud Client Libraries](https://cloud.google.com/apis/docs/cloud-client-libraries) (<https://cloud.google.com/apis/docs/cloud-client-libraries>).

## Prerequisites for this guide

- Understand [service accounts](https://cloud.google.com/iam/docs/service-accounts) (<https://cloud.google.com/iam/docs/service-accounts>)
- Install the [gc1oud tool](https://cloud.google.com/sdk/) (<https://cloud.google.com/sdk/>)

## Required permissions

To allow a user to manage service account keys, grant the *Service Account Key Admin* role (`roles/iam.serviceAccountKeyAdmin`). Cloud IAM primitive roles also contain permissions to manage service account keys, but we recommend granting this role instead to prevent unnecessary access to other Google Cloud resources.

For more information, see the [list of Service Accounts roles](https://cloud.google.com/iam/docs/understanding-roles#service-accounts-roles) (<https://cloud.google.com/iam/docs/understanding-roles#service-accounts-roles>).

## Creating service account keys

To use a service account outside of Google Cloud, such as on other platforms or on-premises, you must first establish the identity of the service account. Public/private key pairs provide a secure way of accomplishing this goal.

You can create a [service account key](#)

(<https://cloud.google.com/iam/reference/rest/v1/projects.serviceAccounts.keys>) using the Cloud Console, the `gc1oud` tool, the `serviceAccounts.keys.create()`.

(<https://cloud.google.com/iam/reference/rest/v1/projects.serviceAccounts.keys/create>) method, or one of the [client libraries](https://cloud.google.com/apis/docs/cloud-client-libraries) (<https://cloud.google.com/apis/docs/cloud-client-libraries>).

In the examples below, **[SA-NAME]** is the name of your service account, and **[PROJECT-ID]** is the ID of your Google Cloud project. You can retrieve the **[SA-NAME]@[PROJECT-ID].iam.gserviceaccount.com** string from the [Service Accounts](#) (<https://console.cloud.google.com/iam-admin/serviceaccounts/>) page in the Cloud Console.

**CONSOLE**      GCLOUD COMMAND      MORE ▾

---

1. Open the **IAM & Admin** page in the Cloud Console.

**OPEN THE IAM & ADMIN PAGE** ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECT/\\_/IAM-ADMIN](https://console.cloud.google.com/project/_/iam-admin))

2. Click **Select a project**, choose a project, and click **Open**.
3. In the left nav, click **Service accounts**.
4. Find the row of the service account that you want to create a key for. In that row, click the **More** **:** button, and then click **Create key**.
5. Select a **Key type** and click **Create**.

Note that the `privateKeyData` returned is a base64-encoded string representation of the `TYPE_GOOGLE_CREDENTIALS_FILE` value (JSON or P12 key/credentials).

When you create a key, your new public/private key pair is generated and downloaded to your machine; it serves as the only copy of the private key. You are responsible for storing the private key securely. Take note of its location and ensure the key is accessible to your application; it needs the key to [make authenticated API calls](#) (<https://developers.google.com/identity/protocols/OAuth2ServiceAccount#callinganapi>).

It may take up to 60 seconds before a newly created key can be used for authentication. If you experience authentication failures immediately after creating a new key, ensure that 60 seconds have elapsed before trying again.

The format of the key may differ depending on how it is generated. Keys created using the Cloud Console or the `gcloud` command-line tool look like this:

```
{
  "type": "service_account",
  "project_id": "[PROJECT-ID]",
  "private_key_id": "[KEY-ID]",
```



```
"private_key": "-----BEGIN PRIVATE KEY-----\n[PRIVATE-KEY]\n-----END PRIVATE KEY-----"
"client_email": "[SERVICE-ACCOUNT-EMAIL]",
"client_id": "[CLIENT-ID]",
"auth_uri": "https://accounts.google.com/o/oauth2/auth",
"token_uri": "https://accounts.google.com/o/oauth2/token",
"auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
"client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/[SERVICE-ACCOUNT-EMAIL].json"
}
```

Whereas keys generated with the REST API or client libraries look like this:

```
{
  "name": "projects/[PROJECT-ID]/serviceAccounts/[SERVICE-ACCOUNT-EMAIL]/keys/[KEY-ID]",
  "privateKeyType": "TYPE_GOOGLE_CREDENTIALS_FILE",
  "privateKeyData": "[PRIVATE-KEY]",
  "validAfterTime": "[DATE]",
  "validBeforeTime": "[DATE]",
  "keyAlgorithm": "KEY_ALG_RSA_2048"
}
```

Note once again that the `privateKeyData` returned is a base64-encoded string representation of the `TYPE_GOOGLE_CREDENTIALS_FILE` value (JSON or P12 key/credentials).

Because the formatting differs between each method, it's easiest to generate a key using the same method you plan to use when making future API calls. For example, if you're using `gcloud`, also generate your key using `gcloud`. To use a key for one method that's been generated using a different method (such as using a REST-generated key with `gcloud`), you'll need to edit the key to match the appropriate format.

Google ensures that all public keys for all service accounts are publicly accessible by anyone and available to verify signatures that are created with the private key. The public key is publicly accessible at the following URLs:

- x.509 certificate: [https://www.googleapis.com/service\\_accounts/v1/metadata/x509/\[SA-NAME\]@\[PROJECT-ID\].iam.gserviceaccount.com](https://www.googleapis.com/service_accounts/v1/metadata/x509/[SA-NAME]@[PROJECT-ID].iam.gserviceaccount.com)
- JSON web key (JWK): [https://www.googleapis.com/service\\_accounts/v1/jwk/\[SA-NAME\]@\[PROJECT-ID\].iam.gserviceaccount.com](https://www.googleapis.com/service_accounts/v1/jwk/[SA-NAME]@[PROJECT-ID].iam.gserviceaccount.com)
- Raw endpoint: [https://www.googleapis.com/service\\_accounts/v1/metadata/raw/\[SA-NAME\]@\[PROJECT-ID\].iam.gserviceaccount.com](https://www.googleapis.com/service_accounts/v1/metadata/raw/[SA-NAME]@[PROJECT-ID].iam.gserviceaccount.com)

## Listing service account keys

You can list the service account keys for a service account using the Cloud Console, the `gcloud` tool, the `serviceAccount.keys.list()`

(<https://cloud.google.com/iam/reference/rest/v1/projects.serviceAccounts.keys/list>) method, or one of the [client libraries](https://cloud.google.com/apis/docs/cloud-client-libraries) (<https://cloud.google.com/apis/docs/cloud-client-libraries>).

The `serviceAccount.keys.list()` method is commonly used to audit service accounts and keys, or to build custom tooling for managing service accounts.

To find out which project your key belongs to, you can download the key as a JSON file and look at that file.

You may see keys listed that you did not create. These are Google Cloud-managed keys used by Google Cloud services such as App Engine and Compute Engine. For more information on the difference between user and Google Cloud-managed keys, see [Understanding service accounts](https://cloud.google.com/iam/docs/understanding-service-accounts) (<https://cloud.google.com/iam/docs/understanding-service-accounts>).

CONSOLE	GCLOUD COMMAND	MORE ▾
<ol style="list-style-type: none"><li>1. Open the <b>IAM &amp; Admin</b> page in the Cloud Console. <b>OPEN THE IAM &amp; ADMIN PAGE</b> (<a href="https://console.cloud.google.com/project/_/iam-admin">HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECT/_/IAM-ADMIN</a>)</li><li>2. Click <b>Select a project</b>, choose a project, and click <b>Open</b>.</li><li>3. In the left nav, click <b>Service accounts</b>. All service accounts and their corresponding keys are listed.</li></ol>		

## Getting a service account key

You can only get the private key data for a service account key when the key is first created.

You can get basic information about a key such as its ID, algorithm, and public key data with the `projects.serviceAccounts.keys.get()`

(<https://cloud.google.com/iam/reference/rest/v1/projects.serviceAccounts.keys/get>) REST API method. Using the Cloud Console or the `gcloud` command-line tool is not supported.

# Uploading public keys for service accounts

## Beta

This feature is in a pre-release state and might change or have limited support. For more information, see the [product launch stages](https://cloud.google.com/products/#product-launch-stages) (<https://cloud.google.com/products/#product-launch-stages>).

You can upload a public key portion of a [user-managed key pair](https://cloud.google.com/iam/docs/service-accounts#user-managed_keys) ([https://cloud.google.com/iam/docs/service-accounts#user-managed\\_keys](https://cloud.google.com/iam/docs/service-accounts#user-managed_keys)) to sign [service account keys](https://cloud.google.com/iam/docs/service-accounts#service_account_keys) ([https://cloud.google.com/iam/docs/service-accounts#service\\_account\\_keys](https://cloud.google.com/iam/docs/service-accounts#service_account_keys)). The public key data is permanently associated with the service account, and will be used for all subsequent signing operations when you create service account keys. If you wish to disable the ability to upload keys for your project, see [restricting service account key upload](https://cloud.google.com/resource-manager/docs/organization-policy/restricting-service-accounts#disable_service_account_key_upload) ([https://cloud.google.com/resource-manager/docs/organization-policy/restricting-service-accounts#disable\\_service\\_account\\_key\\_upload](https://cloud.google.com/resource-manager/docs/organization-policy/restricting-service-accounts#disable_service_account_key_upload)).

If you choose to use a user-managed key pair instead of a Google-managed key pair, ensure that you maintain it and regularly rotate it.

Before uploading your public key, ensure that it is in the [RSA\\_X509\\_PEM](https://cloud.google.com/iot/docs/concepts/device-security#public_key_format) ([https://cloud.google.com/iot/docs/concepts/device-security#public\\_key\\_format](https://cloud.google.com/iot/docs/concepts/device-security#public_key_format)) format. If you do not yet have an existing certificate, you can generate a self-signed X.509 in the appropriate format using tools such as [openssl](https://www.openssl.org/) (<https://www.openssl.org/>). To generate a valid certificate using the `openssl` tool:

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout /output/path/to/private/key \
  -out /output/path/to/public/key -subj "/CN=unused"
```

Note that by default, X.509 certificates created using `openssl` expire after 30 days. However, you can extend or shorten the expiration time using the `-n` flag.

### G CLOUD COMMAND

### REST API

Execute the `gcloud alpha iam service-accounts keys upload` (<https://cloud.google.com/sdk/gcloud/reference/alpha/iam/service-accounts/keys/upload>) command to upload a public key for signing service account keys.

Command:

```
gcloud alpha iam service-accounts keys upload /path/to/public/key \
  --iam-account [SA-NAME]@[PROJECT-ID].iam.gserviceaccount.com
```

The output contains a unique identifier for the uploaded key:

```
Name: projects/PROJECT-ID/serviceAccounts/SA-NAME@PROJECT-ID.iam.gserviceaccount.com
```

To determine whether the command was successful, execute the `gcloud iam service-accounts keys list` (<https://cloud.google.com/sdk/gcloud/reference/iam/service-accounts/keys/list>) command:

```
gcloud iam service-accounts keys list \
  --iam-account [SA-NAME]@[PROJECT-ID].iam.gserviceaccount.com
```

The output will contain the same unique identifier that was returned after the key was created:

KEY_ID	CREATED_AT	EXPIRES_AT
c7b74879da78e4cdcbe7e1bf5e129375c0bfa8d0	2019-06-26T21:01:42.000Z	2029-06-23T21:01:42.000Z

## Deleting service account keys

You can delete a service account key using the Cloud Console, the `gcloud` tool, the `serviceAccount.keys.delete()`

(<https://cloud.google.com/iam/reference/rest/v1/projects.serviceAccounts.keys/delete>) method, or one of the [client libraries](https://cloud.google.com/apis/docs/cloud-client-libraries) (<https://cloud.google.com/apis/docs/cloud-client-libraries>).

If you delete a key, your application will no longer be able to access Cloud Platform resources using that key. A security best practice is to rotate your service account keys regularly. You can rotate a key by creating a new key, switching applications to use the new key and then deleting old key. Use the `serviceAccount.keys.create()` method and `serviceAccount.keys.delete()` method together to automate the rotation.

**CONSOLE**



GCLOUD COMMAND

MORE ▾

1. Open the **IAM & Admin** page in the Cloud Console.

**OPEN THE IAM & ADMIN PAGE** ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECT/\\_/IAM-ADMIN](https://console.cloud.google.com/project/_/iam-admin))

2. Click **Select a project**, choose a project, and click **Open**.
3. In the left nav, click **Service accounts**. All service accounts and their corresponding keys are listed.

4. Find the row of the service account that you want to create a key for. In that row, click the **More**  button in that row, and then click **Create key**.
5. From the list of keys, click **Delete**  for each key you'd like to delete.

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated January 2, 2020.*