Create short-lived credentials for a service account

Archived: 2026-04-05 16:36:28 UTC

Skip to main content

- Technology areas
  - Overview
  - Guides
  - Reference
  - Samples
  - Resources
- Cross-product tools
- Console

- Discover

- Product overview

- Get started

- Grant roles in the Google Cloud console
- Grant roles using client libraries
- IAM and your security architecture
- Identity management for Google Cloud

- Configure identities for users

- Identities for users
- Create and manage Google groups in the Google Cloud console
- Best practices for using Google groups

- Federate identities for users

  - Workforce identity federation
  - SCIM provisioning for Workforce Identity Federation
  -
  -
  - Obtain short-lived credentials for Workforce Identity Federation
  - Manage workforce identity pools and providers
  - Delete Workforce Identity Federation users and their data
  - Set up user access to console (federated)
  - Sign in to the gcloud CLI with your federated identity
  -

- Configure identities for workloads
- Identities for workloads
-
- Use managed workload identities
    - About managed workload identities
    -
    - GKE
        - Create managed workload identities for GKE
        - Troubleshoot managed workload identities for GKE
    - Use custom organization policies
-
-
- Built-in identities for resources
- Control access to resources
-
-
-
- Grant access
    - Manage access to projects, folders, and organizations
    - Manage access to service accounts
    - Manage access to other resources
    - Test allow policy changes
-
- Deny access
-
- Temporary elevated access
    - Temporary elevated access overview
    -
    - Request temporary elevated access with PAM
    - Withdraw grants
    - Approve or deny grants with PAM
    - Create short-lived credentials for a service account
    - Create short-lived credentials for multiple service accounts
    -
    - Migrate to the Service Account Credentials API

- Test permissions for custom user interfaces
- Use custom organization policies for allow policies
- Use IAM to help prevent exfiltration from data pipelines

- Optimize your IAM configuration

- Use IAM securely
- Optimize IAM policies by using Policy Intelligence tools
- Help secure IAM using VPC Service Controls

- Monitor

- Audit logging

    - IAM API audit logging
    - IAM SCIM audit logging
    - Service Account Credentials API audit logging
    - Privileged Access Manager audit logging
    - Security Token Service API audit logging
    - Example logs for service accounts
    - Example logs for Workforce Identity Federation
    - Example logs for Workforce OAuth application integration
    - Example logs for Workload Identity Federation
- Analyze access to resources
-
- Review allow policy history
- Review security insights

- Troubleshoot

-
- Troubleshoot allow and deny policies
- Troubleshoot organization policy errors for service accounts
- Troubleshoot "withcond" in policies and role bindings
- Troubleshoot Workforce Identity Federation
- Troubleshoot Workload Identity Federation

- Samples

- All Identity and Access Management code samples
- Code samples for all products

Create short-lived credentials for a service account Stay organized with collections Save and categorize content based on your preferences.

- On this page

This page explains how to create short-lived credentials for a service account, which you can use to impersonate the service account. Depending on the type of token you create, the short-lived token provides the identity (for ID tokens) or permissions (for access tokens) associated with the service account.

If your system architecture requires you to use a series of token generation calls, you can use a delegation chain consisting of several service accounts. In most cases, the direct method, as explained on this page, is sufficient.

Before you begin

- Enable the IAM and Service Account Credentials APIs:

  Roles required to enable APIs

  To enable APIs, you need the Service Usage Admin IAM role (`roles/serviceusage.serviceUsageAdmin`), which contains the `serviceusage.services.enable` permission. Learn how to grant roles.

  ```
  gcloud services enable iam.googleapis.com iamcredentials.googleapis.com
  ```

- Set up authentication.

  Select the tab for how you plan to use the samples on this page:

  When you use the Google Cloud console to access Google Cloud services and APIs, you don't need to set up authentication.

  In the Google Cloud console, activate Cloud Shell.

  Activate Cloud Shell

  At the bottom of the Google Cloud console, a Cloud Shell session starts and displays a command-line prompt. Cloud Shell is a shell environment with the Google Cloud CLI already installed and with values

already set for your current project. It can take a few seconds for the session to initialize.

To use the Go samples on this page in a local development environment, install and initialize the gcloud CLI, and then set up Application Default Credentials with your user credentials.

1. Install the Google Cloud CLI.
2. If you're using an external identity provider (IdP), you must first sign in to the gcloud CLI with your federated identity.
3. If you're using a local shell, then create local authentication credentials for your user account:

   ```
   gcloud auth application-default login
   ```

   You don't need to do this if you're using Cloud Shell.

   If an authentication error is returned, and you are using an external identity provider (IdP), confirm that you have signed in to the gcloud CLI with your federated identity.

For more information, see Set up ADC for a local development environment in the Google Cloud authentication documentation.

To use the Java samples on this page in a local development environment, install and initialize the gcloud CLI, and then set up Application Default Credentials with your user credentials.

1. Install the Google Cloud CLI.
2. If you're using an external identity provider (IdP), you must first sign in to the gcloud CLI with your federated identity.
3. If you're using a local shell, then create local authentication credentials for your user account:

   ```
   gcloud auth application-default login
   ```

   You don't need to do this if you're using Cloud Shell.

   If an authentication error is returned, and you are using an external identity provider (IdP), confirm that you have signed in to the gcloud CLI with your federated identity.

For more information, see Set up ADC for a local development environment in the Google Cloud authentication documentation.

To use the Node.js samples on this page in a local development environment, install and initialize the gcloud CLI, and then set up Application Default Credentials with your user credentials.

1. Install the Google Cloud CLI.

2. If you're using an external identity provider (IdP), you must first sign in to the gcloud CLI with your federated identity.

3. If you're using a local shell, then create local authentication credentials for your user account:

   ```
   gcloud auth application-default login
   ```

   You don't need to do this if you're using Cloud Shell.

   If an authentication error is returned, and you are using an external identity provider (IdP), confirm that you have signed in to the gcloud CLI with your federated identity.

For more information, see Set up ADC for a local development environment in the Google Cloud authentication documentation.

To use the Python samples on this page in a local development environment, install and initialize the gcloud CLI, and then set up Application Default Credentials with your user credentials.

1. Install the Google Cloud CLI.

2. If you're using an external identity provider (IdP), you must first sign in to the gcloud CLI with your federated identity.

3. If you're using a local shell, then create local authentication credentials for your user account:

   ```
   gcloud auth application-default login
   ```

   You don't need to do this if you're using Cloud Shell.

   If an authentication error is returned, and you are using an external identity provider (IdP), confirm that you have signed in to the gcloud CLI with your federated identity.

For more information, see Set up ADC for a local development environment in the Google Cloud authentication documentation.

To use the REST API samples on this page in a local development environment, you use the credentials you provide to the gcloud CLI.

Install the Google Cloud CLI.

If you're using an external identity provider (IdP), you must first sign in to the gcloud CLI with your federated identity.

For more information, see Authenticate for using REST in the Google Cloud authentication documentation.

- Understand IAM service accounts.

- Understand service account impersonation.
- Understand what kind of token you need, and use the appropriate steps provided in the sections below:
    - OAuth 2.0 access tokens
    - OpenID Connect (OIDC) ID tokens
    - Self-signed JSON Web Tokens (JWTs)
    - Self-signed binary blobs

Create a short-lived access token

Access tokens are accepted for authentication by most Google APIs. When you generate an access token by using service account impersonation, the access token comes without a refresh token, which means that when the token expires, you must repeat the impersonation process to generate a new one.

For more information, see Access tokens.

To create a short-lived access token, complete these tasks:

- Provide the required permissions to the caller.
- Generate the access token.

Provide required permissions

A direct request involves two identities: the caller that requests the credential, and the service account for which the credential is created. How you set up the permissions depends on whether the caller is authenticating as a service account or as a user account.

If you want to run a REST or gcloud CLI command on this page in a local development environment, the caller can be represented by user credentials. For automated workloads, such as an application running on Compute Engine, the caller must be represented by a service account.

When the calling application uses a service account as its identity, the following principals are involved:

- Caller service account (`CALLER_SA`)

  This service account represents the calling application, which issues the request for the short-lived credentials.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To give `CALLER_SA` permissions to create short-lived credentials for `PRIV_SA` , you grant `CALLER_SA` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA`.
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.
6. Enter the email address of the caller service account, `CALLER_SA`.

   For example, `demo@my-project.iam.gserviceaccount.com`.
7. Select the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).
8. Click **Save** to grant the role to the service account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- `PRIV_SA`: The email address of the privilege-bearing service account for which the token is generated.
- `CALLER_SA`: The email address of the service account representing the application that is requesting the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
        "serviceAccount:CALLER_SA"
      ],
      "role": "roles/iam.serviceAccountTokenCreator"
    }
  ],
  "etag": "BwXhCB4eyjY=",
  "version": 1
}
```

1. Read the allow policy for `PRIV_SA`:

   The `serviceAccounts.getIamPolicy` method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - `PROJECT_ID`: Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project`.
   - `PRIV_SA`: The email address of the privilege-bearing service account for which the short-lived token is created.
   - `POLICY_VERSION`: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant `CALLER_SA` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - `PROJECT_ID`: Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project`.
   - `PRIV_SA`: The email address of the privilege-bearing service account for which the short-lived token is created.
   - `POLICY_VERSION`: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - `POLICY`: A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace `POLICY` with the following, where `CALLER_SA` is the service account creating the short-lived token:

```
{
  "version": 1,
```

```
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

When you want to use the Google Cloud CLI to generate short-lived tokens, or you want to generate short-lived tokens from a local development environment, you can use a user account to generate the tokens. Often, you can use your own user account.

When you use a user account to generate short-lived tokens, the following identities are involved:

- Caller account (`CALLER_ACCOUNT`)

  This user account is used to generate short-lived credentials for the privilege-bearing service account.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To enable `CALLER_ACCOUNT` to create short-lived credentials for `PRIV_SA` , you grant `CALLER_ACCOUNT` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA` .
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.

6. Enter the principal identifier of the caller account, `CALLER_ACCOUNT`.

   For example, `my-user@example.com`.

7. Select the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).
8. Click **Save** to grant the role to the user account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- `PRIV_SA`: The email address of the privilege-bearing service account for which the token is generated.
- `CALLER_ACCOUNT`: The email address of the user account being used to request the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
        "user:my-user@example.com"
      ],
      "role": "roles/iam.serviceAccountTokenCreator"
    }
  ],
  "etag": "BwX1ZbefjXU=",
```

```
  "version": 1
}
```

1. Read the allow policy for *PRIV_SA*:

   The serviceAccounts.getIamPolicy method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - *PROJECT_ID*: Your Google Cloud project ID. Project IDs are alphanumeric strings, like my-project.
   - *PRIV_SA*: The email address of the privilege-bearing service account for which the short-lived token is created.
   - *POLICY_VERSION*: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named request.json, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
        -H "Content-Type: application/json; charset=utf-8" \
        -d @request.json \
        "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
   ```

   PowerShell (Windows)

   Save the request body in a file named request.json, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/
```

PROJECT_ID/serviceAccounts/*PRIV_SA*:getIamPolicy" | Select-Object -Expand Content

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant *`CALLER_ACCOUNT`* the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
```

```
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - *`PROJECT_ID`* : Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project` .
   - *`PRIV_SA`* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *`POLICY_VERSION`* : The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - *`POLICY`* : A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace *`POLICY`* with the following, where *`CALLER_ACCOUNT`* is the user account creating the short-lived token:

     ```
     {
       "version": 1,
       "etag": "BwWKmjvelug=",
       "bindings": [
         {
           "role": "roles/serviceAccountAdmin",
           "members": [
             "user:my-user@example.com"
           ]
         },
         {
           "role": "roles/iam.serviceAccountTokenCreator",
     ```

```
      "members": [
        "CALLER_ACCOUNT"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

Generate the access token

You can generate an OAuth 2.0 access token by using the gcloud CLI, the REST API, or the Cloud Client Libraries and Google API Client Libraries.

If you use the REST API, and your system is configured to allow extended token lifetimes, you can create a token with a lifetime longer than the default. The Google Cloud CLI does not support setting a lifetime for the token.

The samples below are designed to be used in a local development environment; the caller must be represented by a user account, rather than a service account.

Generate an OAuth 2.0 access token for a service account:

1. Ensure that you are signed into the gcloud CLI with the caller user account.
2. Generate a token for the service account by using the `gcloud auth print-access-token` command.

   Before using any of the command data below, make the following replacements:

   - *PRIV_SA* : The email address of the privilege-bearing service account for which the short-lived token is created.

   Execute the following command:

   Linux, macOS, or Cloud Shell

```
gcloud auth print-access-token --impersonate-service-account=PRIV_SA
```

Windows (PowerShell)

```
gcloud auth print-access-token --impersonate-service-account=PRIV_SA
```

Windows (cmd.exe)

```
gcloud auth print-access-token --impersonate-service-account=PRIV_SA
```

You should receive a response similar to the following:

```
WARNING: This command is using service account impersonation. All API calls will be executed a
[my-sa@my-project.iam.gserviceaccount.com].
ya29.c.b0AXv0zTPnzTnDV8F8Aj5Fgy46Yf2v_v8eZIoKq7xGpfbpXuy23aQ1693m3gAuE8AZga7w6kdagN7a9bfdDYbde
```

The Service Account Credentials API's `serviceAccounts.generateAccessToken` method generates an OAuth 2.0 access token for a service account.

Before using any of the request data, make the following replacements:

- *PRIV_SA* : The email address of the privilege-bearing service account for which the short-lived token is created.
- *LIFETIME* : The amount of time until the access token expires, in seconds. For example, `300s` .

  By default, the maximum token lifetime is 1 hour (3,600 seconds). To extend the maximum lifetime for these tokens to 12 hours (43,200 seconds), add the service account to an organization policy that includes the `constraints/iam.allowServiceAccountCredentialLifetimeExtension` list constraint.

HTTP method and URL:

```
POST https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:generateAccessToken
```

Request JSON body:

```
{
  "scope": [
    "https://www.googleapis.com/auth/cloud-platform"
  ],
  "lifetime": "LIFETIME"
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:generateAccessToke
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/

PRIV_SA:generateAccessToken" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

If the `generateAccessToken` request was successful, the response body contains an OAuth 2.0 access token and an expiration time. The `accessToken` can then be used to authenticate a request on behalf of the service account until the `expireTime` has been reached:

```
{
  "accessToken": "eyJ0eXAi...NiJ9",
  "expireTime": "2020-04-07T15:01:23.045123456Z"
}
```

Create an OpenID Connect (OIDC) ID token

ID tokens follow the OpenID Connect (OIDC) specification. ID tokens are accepted by a limited number of services and applications.

For more information, see ID tokens and Authentication for applications hosted on Cloud Run or Cloud Run functions.

To create an ID token, complete these tasks:

- Provide the required permissions to the caller.

  Use the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`) for creating an ID token. This is a different role than the role you use for other token types.

- Generate the ID token.

Provide required permissions

A direct request involves two identities: the caller that requests the credential, and the service account for which the credential is created. How you set up the permissions depends on whether the caller is authenticating as a service account or as a user account.

If you want to run a REST or gcloud CLI command on this page in a local development environment, the caller can be represented by user credentials. For automated workloads, such as an application running on Compute Engine, the caller must be represented by a service account.

When the calling application uses a service account as its identity, the following principals are involved:

- Caller service account (*`CALLER_SA`*)

  This service account represents the calling application, which issues the request for the short-lived credentials.

- Privilege-bearing service account (*`PRIV_SA`*)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To give *`CALLER_SA`* permissions to create short-lived credentials for *`PRIV_SA`* , you grant *`CALLER_SA`* the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`) on *`PRIV_SA`* .

Grant the required role on *`PRIV_SA`* :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.

3. Click the email address of the privilege-bearing service account, `PRIV_SA`.
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.
6. Enter the email address of the caller service account, `CALLER_SA`.

   For example, `demo@my-project.iam.gserviceaccount.com`.
7. Select the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`).
8. Click **Save** to grant the role to the service account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- `PRIV_SA`: The email address of the privilege-bearing service account for which the token is generated.
- `CALLER_SA`: The email address of the service account representing the application that is requesting the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountOpenIdTokenCreator --format=jso
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountOpenIdTokenCreator --format=jso
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountOpenIdTokenCreator --format=jso
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
```

```
        "serviceAccount:CALLER_SA"
      ],
      "role": "roles/iam.serviceAccountOpenIdTokenCreator"
    }
  ],
  "etag": "BwXhCB4eyjY=",
  "version": 1
}
```

1. Read the allow policy for `PRIV_SA`:

   The `serviceAccounts.getIamPolicy` method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - `PROJECT_ID`: Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project`.
   - `PRIV_SA`: The email address of the privilege-bearing service account for which the short-lived token is created.
   - `POLICY_VERSION`: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named `request.json`, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
        -H "Content-Type: application/json; charset=utf-8" \
   ```

```
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant *`CALLER_SA`* the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`).

For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountOpenIdTokenCreator",
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - *PROJECT_ID* : Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project` .
   - *PRIV_SA* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *POLICY_VERSION* : The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - *POLICY* : A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace *POLICY* with the following, where *CALLER_SA* is the service account creating the short-lived token:

     ```
     {
       "version": 1,
       "etag": "BwWKmjvelug=",
       "bindings": [
         {
     ```

```
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountOpenIdTokenCreator",
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }
```

```
Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

When you want to use the Google Cloud CLI to generate short-lived tokens, or you want to generate short-lived tokens from a local development environment, you can use a user account to generate the tokens. Often, you can use your own user account.

When you use a user account to generate short-lived tokens, the following identities are involved:

- Caller account (`CALLER_ACCOUNT`)

  This user account is used to generate short-lived credentials for the privilege-bearing service account.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To enable `CALLER_ACCOUNT` to create short-lived credentials for `PRIV_SA` , you grant `CALLER_ACCOUNT` the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA` .
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.
6. Enter the principal identifier of the caller account, `CALLER_ACCOUNT` .

For example, `my-user@example.com`.

7. Select the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`).
8. Click **Save** to grant the role to the user account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- *`PRIV_SA`*: The email address of the privilege-bearing service account for which the token is generated.
- *`CALLER_ACCOUNT`*: The email address of the user account being used to request the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountOpenIdTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountOpenIdTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountOpenIdTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
        "user:my-user@example.com"
      ],
      "role": "roles/iam.serviceAccountOpenIdTokenCreator"
    }
  ],
  "etag": "BwX1ZbefjXU=",
```

```
  "version": 1
}
```

1. Read the allow policy for *PRIV_SA*:

   The serviceAccounts.getIamPolicy method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - *PROJECT_ID*: Your Google Cloud project ID. Project IDs are alphanumeric strings, like my-project.
   - *PRIV_SA*: The email address of the privilege-bearing service account for which the short-lived token is created.
   - *POLICY_VERSION*: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named request.json, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
        -H "Content-Type: application/json; charset=utf-8" \
        -d @request.json \
        "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
   ```

   PowerShell (Windows)

   Save the request body in a file named request.json, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant *`CALLER_ACCOUNT`* the Service Account OpenID Connect Identity Token Creator role (`roles/iam.serviceAccountOpenIdTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
```

```
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountOpenIdTokenCreator",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - *`PROJECT_ID`* : Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project` .
   - *`PRIV_SA`* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *`POLICY_VERSION`* : The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - *`POLICY`* : A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace *`POLICY`* with the following, where *`CALLER_ACCOUNT`* is the user account creating the short-lived token:

     ```
     {
       "version": 1,
       "etag": "BwWKmjvelug=",
       "bindings": [
         {
           "role": "roles/serviceAccountAdmin",
           "members": [
             "user:my-user@example.com"
           ]
         },
         {
           "role": "roles/iam.serviceAccountOpenIdTokenCreator",
     ```

```
      "members": [
        "CALLER_ACCOUNT"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

Generate the ID token

You can generate an OpenID Connect (OIDC) ID token by using the gcloud CLI, the REST API, or the Cloud Client Libraries and Google API Client Libraries.

The samples below are designed to be used in a local development environment; the caller must be represented by a user account, rather than a service account.

OIDC ID tokens are valid for 1 hour (3,600 seconds).

Generate a Google-signed OIDC ID token for a service account:

1. Ensure that you are signed into the gcloud CLI with the caller user account.
2. Generate a token for the service account by using the `gcloud auth print-identity-token` command.

   Before using any of the command data below, make the following replacements:

   - *`PRIV_SA`* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *`AUDIENCE_NAME`* : The audience for the token, usually the URL of the application or service that the token will be used to access.

     Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud auth print-identity-token --impersonate-service-account=PRIV_SA --audience
```

Windows (PowerShell)

```
gcloud auth print-identity-token --impersonate-service-account=PRIV_SA --audience
```

Windows (cmd.exe)

```
gcloud auth print-identity-token --impersonate-service-account=PRIV_SA --audience
```

You should receive a response similar to the following:

```
WARNING: This command is using service account impersonation. All API calls will
[my-sa@my-project.iam.gserviceaccount.com].
eyJhbGciOiJSUzI1NiIsImtpZDNhMDg4ZDRmZmMjJkYTVmZTM5MDZjY2MiLCJ0eXAiOiJKV1QifQ.eyJh
```

The Service Account Credentials API's `serviceAccounts.generateIdToken` method generates an OIDC ID token for a service account.

Before using any of the request data, make the following replacements:

- `PRIV_SA` : The email address of the privilege-bearing service account for which the short-lived token is created.
- `AUDIENCE_NAME` : The audience for the token, usually the URL of the application or service that the token will be used to access.

HTTP method and URL:

```
POST https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:generateIdToken
```

Request JSON body:

```
{
  "audience": "AUDIENCE_NAME",
  "includeEmail": "true"
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:generateIdToken"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/

PRIV_SA:generateIdToken" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

If the `generateId` request was successful, the response body contains an ID token that is valid for 1 hour. The `token` can then be used to authenticate a request on behalf of the service account:

```
{
  "token": "eyJ0eXAi...NiJ9"
}
```

Create a self-signed JSON Web Token (JWT)

Self-signed JSON Web Tokens (JWTs) are useful in a variety of scenarios:

- Securely communicating between your own applications. In this scenario, one application can sign a token that can be verified by another application for authentication purposes.

- Authenticating a call to a Google API as described in Service account authorization without OAuth.
- Authenticating to an API deployed with API Gateway.
- Treating a service account as an identity provider by signing a JWT that contains arbitrary claims about a user, account, or device.

To create a JWT, complete these tasks:

- Provide the required permissions to the caller.
- Generate the JWT.

Provide required permissions

A direct request involves two identities: the caller that requests the credential, and the service account for which the credential is created. How you set up the permissions depends on whether the caller is authenticating as a service account or as a user account.

If you want to run a REST or gcloud CLI command on this page in a local development environment, the caller can be represented by user credentials. For automated workloads, such as an application running on Compute Engine, the caller must be represented by a service account.

When the calling application uses a service account as its identity, the following principals are involved:

- Caller service account (`CALLER_SA`)

  This service account represents the calling application, which issues the request for the short-lived credentials.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To give `CALLER_SA` permissions to create short-lived credentials for `PRIV_SA` , you grant `CALLER_SA` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA` .
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.
6. Enter the email address of the caller service account, `CALLER_SA` .

For example, `demo@my-project.iam.gserviceaccount.com`.

7. Select the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).
8. Click **Save** to grant the role to the service account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- *`PRIV_SA`*: The email address of the privilege-bearing service account for which the token is generated.
- *`CALLER_SA`*: The email address of the service account representing the application that is requesting the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
        "serviceAccount:CALLER_SA"
      ],
      "role": "roles/iam.serviceAccountTokenCreator"
    }
  ],
  "etag": "BwXhCB4eyjY=",
```

```
  "version": 1
}
```

1. Read the allow policy for `PRIV_SA`:

   The `serviceAccounts.getIamPolicy` method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - `PROJECT_ID`: Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project`.
   - `PRIV_SA`: The email address of the privilege-bearing service account for which the short-lived token is created.
   - `POLICY_VERSION`: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named `request.json`, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
        -H "Content-Type: application/json; charset=utf-8" \
        -d @request.json \
        "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
   ```

   PowerShell (Windows)

   Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/
```

PROJECT_ID/serviceAccounts/*PRIV_SA*:getIamPolicy" | Select-Object -Expand Content

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant *`CALLER_SA`* the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
  {
    "version": 1,
    "etag": "BwWKmjvelug=",
    "bindings": [
      {
```

```
        "role": "roles/serviceAccountAdmin",
        "members": [
          "user:my-user@example.com"
        ]
      },
      {
        "role": "roles/iam.serviceAccountTokenCreator",
        "members": [
          "serviceAccount:CALLER_SA"
        ]
      }
    ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - *`PROJECT_ID`* : Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project` .
   - *`PRIV_SA`* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *`POLICY_VERSION`* : The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - *`POLICY`* : A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace *`POLICY`* with the following, where *`CALLER_SA`* is the service account creating the short-lived token:

     ```
     {
       "version": 1,
       "etag": "BwWKmjvelug=",
       "bindings": [
         {
           "role": "roles/serviceAccountAdmin",
           "members": [
             "user:my-user@example.com"
           ]
         },
         {
           "role": "roles/iam.serviceAccountTokenCreator",
     ```

```
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

When you want to use the Google Cloud CLI to generate short-lived tokens, or you want to generate short-lived tokens from a local development environment, you can use a user account to generate the tokens. Often, you can use your own user account.

When you use a user account to generate short-lived tokens, the following identities are involved:

- Caller account (`CALLER_ACCOUNT`)

  This user account is used to generate short-lived credentials for the privilege-bearing service account.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To enable `CALLER_ACCOUNT` to create short-lived credentials for `PRIV_SA` , you grant `CALLER_ACCOUNT` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA` .
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.

6. Enter the principal identifier of the caller account, `CALLER_ACCOUNT`.

   For example, `my-user@example.com`.

7. Select the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).
8. Click **Save** to grant the role to the user account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- `PRIV_SA` : The email address of the privilege-bearing service account for which the token is generated.
- `CALLER_ACCOUNT` : The email address of the user account being used to request the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
        "user:my-user@example.com"
      ],
      "role": "roles/iam.serviceAccountTokenCreator"
    }
  ],
  "etag": "BwX1ZbefjXU=",
```

```
  "version": 1
}
```

1. Read the allow policy for *PRIV_SA*:

   The serviceAccounts.getIamPolicy method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - *PROJECT_ID*: Your Google Cloud project ID. Project IDs are alphanumeric strings, like my-project.
   - *PRIV_SA*: The email address of the privilege-bearing service account for which the short-lived token is created.
   - *POLICY_VERSION*: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named request.json, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
        -H "Content-Type: application/json; charset=utf-8" \
        -d @request.json \
        "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
   ```

   PowerShell (Windows)

   Save the request body in a file named request.json, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/
```

PROJECT_ID/serviceAccounts/*PRIV_SA*:getIamPolicy" | Select-Object -Expand Content

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant *`CALLER_ACCOUNT`* the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
```

```
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - *`PROJECT_ID`* : Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project` .
   - *`PRIV_SA`* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *`POLICY_VERSION`* : The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - *`POLICY`* : A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace *`POLICY`* with the following, where *`CALLER_ACCOUNT`* is the user account creating the short-lived token:

     ```
     {
       "version": 1,
       "etag": "BwWKmjvelug=",
       "bindings": [
         {
           "role": "roles/serviceAccountAdmin",
           "members": [
             "user:my-user@example.com"
           ]
         },
         {
           "role": "roles/iam.serviceAccountTokenCreator",
     ```

```
      "members": [
        "CALLER_ACCOUNT"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

Generate the JWT

Generate a self-signed JWT:

The Service Account Credentials API's `serviceAccounts.signJwt` method signs a JWT using a service account's system-managed private key.

Before using any of the request data, make the following replacements:

- *PRIV_SA* : The email address of the privilege-bearing service account for which the short-lived token is created.
- *JWT_PAYLOAD* : The JWT payload to sign, which is a JSON object that contains a JWT Claims Set. Include the claims that are necessary for your desired use case and to meet the validation requirements for the service you are calling. If you are calling a Google API, see Google's Authentication Guide for claim requirements.

  The `exp` (expiration time) claim must be no more than 12 hours in the future. If you are calling a Google API, the `exp` claim must be set no more than 1 hour in the future.

  The following example payload contains claims to call a Google API, where *EXP* is an integer timestamp representing the expiration time:

```
{ \"iss\": \"PRIV_SA\", \"sub\": \"PRIV_SA\", \"aud\": \"https://firestore.googleapis.com/\",
```

HTTP method and URL:

```
POST https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:signJwt
```

Request JSON body:

```
{
  "payload": "JWT_PAYLOAD"
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:signJwt"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/

PRIV_SA:signJwt" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

If the `signJwt` request was successful, the response body contains a signed JWT and the signing key ID that was used to sign the JWT. You can use the `signedJwt` value as a bearer token to directly authenticate a request on

behalf of the service account. The token is valid up to the expiration time specified in the request:

```
{
  "keyId": "42ba1e...fc0a",
  "signedJwt": "eyJ0eXAi...NiJ9"
}
```

Create a self-signed binary object (blob)

Self-signed binary objects, or blobs, are used to transmit binary data in such a way that the originator of the data is known (because the blob is self-signed). Blobs can be used to create signatures, a Cloud Storage object required for various authentication flows including signed URLs. For information about signatures, see the Cloud Storage documentation.

To create a self-signed binary object, complete these tasks:

- Provide the required permissions to the caller.
- Generate the self-signed blob.

Provide required permissions

A direct request involves two identities: the caller that requests the credential, and the service account for which the credential is created. How you set up the permissions depends on whether the caller is authenticating as a service account or as a user account.

If you want to run a REST or gcloud CLI command on this page in a local development environment, the caller can be represented by user credentials. For automated workloads, such as an application running on Compute Engine, the caller must be represented by a service account.

When the calling application uses a service account as its identity, the following principals are involved:

- Caller service account (`CALLER_SA`)

  This service account represents the calling application, which issues the request for the short-lived credentials.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To give `CALLER_SA` permissions to create short-lived credentials for `PRIV_SA` , you grant `CALLER_SA` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA`.
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.
6. Enter the email address of the caller service account, `CALLER_SA`.

   For example, `demo@my-project.iam.gserviceaccount.com`.
7. Select the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).
8. Click **Save** to grant the role to the service account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- `PRIV_SA`: The email address of the privilege-bearing service account for which the token is generated.
- `CALLER_SA`: The email address of the service account representing the application that is requesting the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=serviceAccount:CALLER_SA --role=roles/iam.serviceAccountTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
```

```
    {
      "members": [
        "serviceAccount:CALLER_SA"
      ],
      "role": "roles/iam.serviceAccountTokenCreator"
    }
  ],
  "etag": "BwXhCB4eyjY=",
  "version": 1
}
```

1. Read the allow policy for `PRIV_SA`:

   The `serviceAccounts.getIamPolicy` method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - `PROJECT_ID`: Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project`.
   - `PRIV_SA`: The email address of the privilege-bearing service account for which the short-lived token is created.
   - `POLICY_VERSION`: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named `request.json`, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
   ```

```
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant `CALLER_SA` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - `PROJECT_ID`: Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project`.
   - `PRIV_SA`: The email address of the privilege-bearing service account for which the short-lived token is created.
   - `POLICY_VERSION`: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - `POLICY`: A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace `POLICY` with the following, where `CALLER_SA` is the service account creating the short-lived token:

```
{
  "version": 1,
```

```
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "serviceAccount:CALLER_SA"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

When you want to use the Google Cloud CLI to generate short-lived tokens, or you want to generate short-lived tokens from a local development environment, you can use a user account to generate the tokens. Often, you can use your own user account.

When you use a user account to generate short-lived tokens, the following identities are involved:

- Caller account (`CALLER_ACCOUNT`)

  This user account is used to generate short-lived credentials for the privilege-bearing service account.

- Privilege-bearing service account (`PRIV_SA`)

  This service account is granted the IAM roles needed for the short-lived token. This is the service account for which the short-lived token is created.

To enable `CALLER_ACCOUNT` to create short-lived credentials for `PRIV_SA` , you grant `CALLER_ACCOUNT` the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`) on `PRIV_SA` .

Grant the required role on `PRIV_SA` :

1. In the Google Cloud console, go to the **Service Accounts** page.

   Go to Service Accounts

2. Select a project.
3. Click the email address of the privilege-bearing service account, `PRIV_SA` .
4. Click the **Permissions** tab.
5. Under **Principals with access to this service account**, click **Grant Access**.

6. Enter the principal identifier of the caller account, `CALLER_ACCOUNT`.

   For example, `my-user@example.com`.

7. Select the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).
8. Click **Save** to grant the role to the user account.

The `gcloud iam service-accounts add-iam-policy-binding` command grants a role on a service account.

Before using any of the command data below, make the following replacements:

- `PRIV_SA`: The email address of the privilege-bearing service account for which the token is generated.
- `CALLER_ACCOUNT`: The email address of the user account being used to request the short-lived token.

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA \
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (PowerShell)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA `
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

Windows (cmd.exe)

```
gcloud iam service-accounts add-iam-policy-binding PRIV_SA ^
    --member=user:CALLER_ACCOUNT --role=roles/iam.serviceAccountTokenCreator --format=json
```

You should receive a response similar to the following:

```
Updated IAM policy for serviceAccount [PRIV_SA].
{
  "bindings": [
    {
      "members": [
        "user:my-user@example.com"
      ],
      "role": "roles/iam.serviceAccountTokenCreator"
    }
  ],
  "etag": "BwX1ZbefjXU=",
```

```
  "version": 1
}
```

1. Read the allow policy for *PRIV_SA*:

   The serviceAccounts.getIamPolicy method gets a service account's allow policy.

   Before using any of the request data, make the following replacements:

   - *PROJECT_ID*: Your Google Cloud project ID. Project IDs are alphanumeric strings, like my-project.
   - *PRIV_SA*: The email address of the privilege-bearing service account for which the short-lived token is created.
   - *POLICY_VERSION*: The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.

   HTTP method and URL:

   ```
   POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy
   ```

   Request JSON body:

   ```
   {
     "options": {
       "requestedPolicyVersion": POLICY_VERSION
     }
   }
   ```

   To send your request, expand one of these options:

   curl (Linux, macOS, or Cloud Shell)

   Save the request body in a file named request.json, and execute the following command:

   ```
   curl -X POST \
        -H "Authorization: Bearer $(gcloud auth print-access-token)" \
        -H "Content-Type: application/json; charset=utf-8" \
        -d @request.json \
        "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:getIamPolicy"
   ```

   PowerShell (Windows)

   Save the request body in a file named request.json, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/
```

PROJECT_ID/serviceAccounts/*PRIV_SA*:getIamPolicy" | Select-Object -Expand Content

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

You should receive a JSON response similar to the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

If you have not granted any roles on the service account, the response contains only an `etag` value. Include that `etag` value in the next step.

2. Modify the allow policy to grant *`CALLER_ACCOUNT`* the Service Account Token Creator role (`roles/iam.serviceAccountTokenCreator`).

   For example, to modify the sample response from the previous step, add the following:

```
{
  "version": 1,
  "etag": "BwWKmjvelug=",
  "bindings": [
    {
```

```
      "role": "roles/serviceAccountAdmin",
      "members": [
        "user:my-user@example.com"
      ]
    },
    {
      "role": "roles/iam.serviceAccountTokenCreator",
      "members": [
        "user:my-user@example.com"
      ]
    }
  ]
}
```

3. Write the updated allow policy:

   The `serviceAccounts.setIamPolicy` method sets an updated allow policy for the service account.

   Before using any of the request data, make the following replacements:

   - *`PROJECT_ID`* : Your Google Cloud project ID. Project IDs are alphanumeric strings, like `my-project` .
   - *`PRIV_SA`* : The email address of the privilege-bearing service account for which the short-lived token is created.
   - *`POLICY_VERSION`* : The policy version to be returned. Requests should specify the most recent policy version, which is policy version 3. See Specifying a policy version when getting a policy for details.
   - *`POLICY`* : A JSON representation of the policy that you want to set. For more information about the format of a policy, see the Policy reference.

     For example, to set the allow policy shown in the previous step, replace *`POLICY`* with the following, where *`CALLER_ACCOUNT`* is the user account creating the short-lived token:

     ```
     {
       "version": 1,
       "etag": "BwWKmjvelug=",
       "bindings": [
         {
           "role": "roles/serviceAccountAdmin",
           "members": [
             "user:my-user@example.com"
           ]
         },
         {
           "role": "roles/iam.serviceAccountTokenCreator",
     ```

```
      "members": [
        "CALLER_ACCOUNT"
      ]
    }
  ]
}
```

HTTP method and URL:

```
POST https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy
```

Request JSON body:

```
{
  "policy": POLICY
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iam.googleapis.com/v1/projects/PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iam.googleapis.com/v1/projects/

PROJECT_ID/serviceAccounts/PRIV_SA:setIamPolicy" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

The response contains the updated allow policy.

Generate the self-signed blob

Generate a self-signed blob for the service account:

The Service Account Credentials API's `serviceAccounts.signBlob` method signs a blob using a service account's system-managed private key.

Before using any of the request data, make the following replacements:

- *PRIV_SA* : The email address of the privilege-bearing service account for which the short-lived token is created.
- *BLOB_PAYLOAD* : A base64-encoded string of bytes. For example, `VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wZWQgb3ZlciB0aGUgbGF6eSBkb2cu` .

HTTP method and URL:

```
POST https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:signBlob
```

Request JSON body:

```
{
  "payload": "BLOB_PAYLOAD"
}
```

To send your request, expand one of these options:

curl (Linux, macOS, or Cloud Shell)

Save the request body in a file named `request.json`, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/PRIV_SA:signBlob"
```

PowerShell (Windows)

Save the request body in a file named `request.json`, and execute the following command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/

PRIV_SA:signBlob" | Select-Object -Expand Content
```

APIs Explorer (browser)

Copy the request body and open the method reference page. The APIs Explorer panel opens on the right side of the page. You can interact with this tool to send requests. Paste the request body in this tool, complete any other required fields, and click **Execute**.

If the `signBlob` request was successful, the response body contains a signed blob and the signing key ID that was used to sign the blob. You can use the `signedBlob` value as a bearer token to directly authenticate a request on behalf of the service account. The token is valid until the service account's system-managed private key expires. This key's ID is the value of the `keyId` field in the response.

```
{
  "keyId": "42ba1e...fc0a",
  "signedBlob": "eyJ0eXAi...NiJ9"
}
```

Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2026-04-02 UTC.
