List secrets and view secret details

Archived: 2026-04-06 01:03:25 UTC

In Secret Manager, a secret acts as a container for multiple secret versions. It holds important information like labels and rotation settings, but not the secret value itself. This page explains how to retrieve a list of all secrets within a project and view the metadata associated with each secret.

Required roles

To get the permissions that you need to list secrets and view secret metadata, ask your administrator to grant you the Secret Manager Viewer (`roles/secretmanager.viewer`) IAM role on the project, folder, or organization. For more information about granting roles, see Manage access to projects, folders, and organizations.

You might also be able to get the required permissions through custom roles or other predefined roles.

To retrieve a list of all secrets within a project, use one of the following methods:

1. In the Google Cloud console, go to the **Secret Manager** page.

   Go to Secret Manager

2. Check the list of secrets in the project. You can click a secret to view the secret metadata.

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

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud secrets list
```

Windows (PowerShell)

```
gcloud secrets list
```

Windows (cmd.exe)

```
gcloud secrets list
```

The response returns the list of secrets and their secret versions.

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

- *PROJECT_ID*: the Google Cloud project ID

HTTP method and URL:

```
GET https://secretmanager.googleapis.com/v1/projects/PROJECT_ID/secrets
```

Request JSON body:

```
{}
```

To send your request, choose one of these options:

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

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

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 GET `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://secretmanager.googleapis.com/v1/projects/

PROJECT_ID/secrets" | Select-Object -Expand Content
```

You should receive a JSON response similar to the following:

```
{
  "secrets": [
    {
      "name": "projects/PROJECT_ID/locations/LOCATION/secrets/SECRET_ID",
      "createTime": "2024-09-02T07:14:00.281541Z",
      "etag": "\"16211daf5f29c5\""
    },
  ],
  "totalSize": 1
```

```
}
```

To run this code, first set up a C# development environment and install the Secret Manager C# SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Go development environment and install the Secret Manager Go SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Java development environment and install the Secret Manager Java SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Node.js development environment and install the Secret Manager Node.js SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first learn about using PHP on Google Cloud and install the Secret Manager PHP SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Python development environment and install the Secret Manager Python SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Ruby development environment and install the Secret Manager Ruby SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

View secret details

To view a secret's metadata, use one of the following methods:

1. In the Google Cloud console, go to the **Secret Manager** page.

   Go to Secret Manager

2. Click the secret whose details you want to view.

3. On the secret details page, click the **Overview** tab. This tab displays the general details and metadata associated with the secret.

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

- *SECRET_ID*: the ID of the secret

Execute the following command:

Linux, macOS, or Cloud Shell

```
gcloud secrets describe SECRET_ID
```

Windows (PowerShell)

```
gcloud secrets describe SECRET_ID
```

Windows (cmd.exe)

```
gcloud secrets describe SECRET_ID
```

The response returns the secret.

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

- *PROJECT_ID*: the Google Cloud project ID
- *SECRET_ID*: the ID of the secret

HTTP method and URL:

```
GET https://secretmanager.googleapis.com/v1/projects/PROJECT_ID/secrets/SECRET_ID
```

Request JSON body:

```
{}
```

To send your request, choose one of these options:

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

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

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 GET `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
```

```
    -InFile request.json `
    -Uri "https://secretmanager.googleapis.com/v1/projects/

PROJECT_ID/secrets/SECRET_ID" | Select-Object -Expand Content
```

You should receive a JSON response similar to the following:

```
{
  "name": "projects/PROJECT_ID/locations/LOCATION/secrets/SECRET_ID",
  "createTime": "2024-09-02T07:14:00.281541Z",
  "etag": "\"16211daf5f29c5\""
}
```

To run this code, first set up a C# development environment and install the Secret Manager C# SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Go development environment and install the Secret Manager Go SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Java development environment and install the Secret Manager Java SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Node.js development environment and install the Secret Manager Node.js SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first learn about using PHP on Google Cloud and install the Secret Manager PHP SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Python development environment and install the Secret Manager Python SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

To run this code, first set up a Ruby development environment and install the Secret Manager Ruby SDK. On Compute Engine or GKE, you must authenticate with the `cloud-platform` scope.

What's next

- Learn how to edit secrets.
- Learn how to set up rotation schedules for secrets.
- Learn how to set up notifications on a secret.
